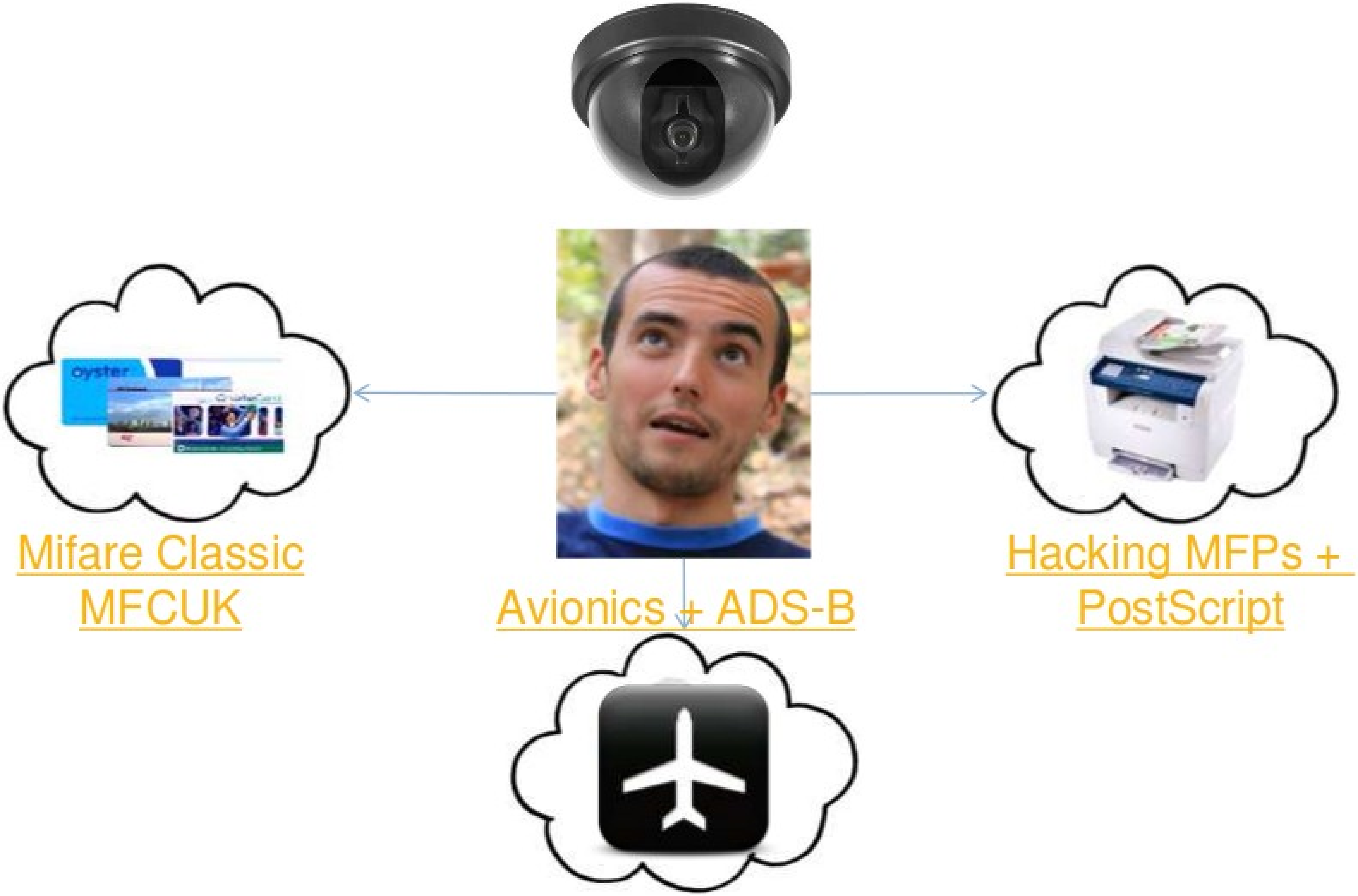
Security of Embedded Devices' Firmware: Fast and Furious at Large Scale

Andrei Costin, PhD
www.firmware.re

firmware · ǝɹ

# whoami

- Embedded security researcher,fresh Dr. :)



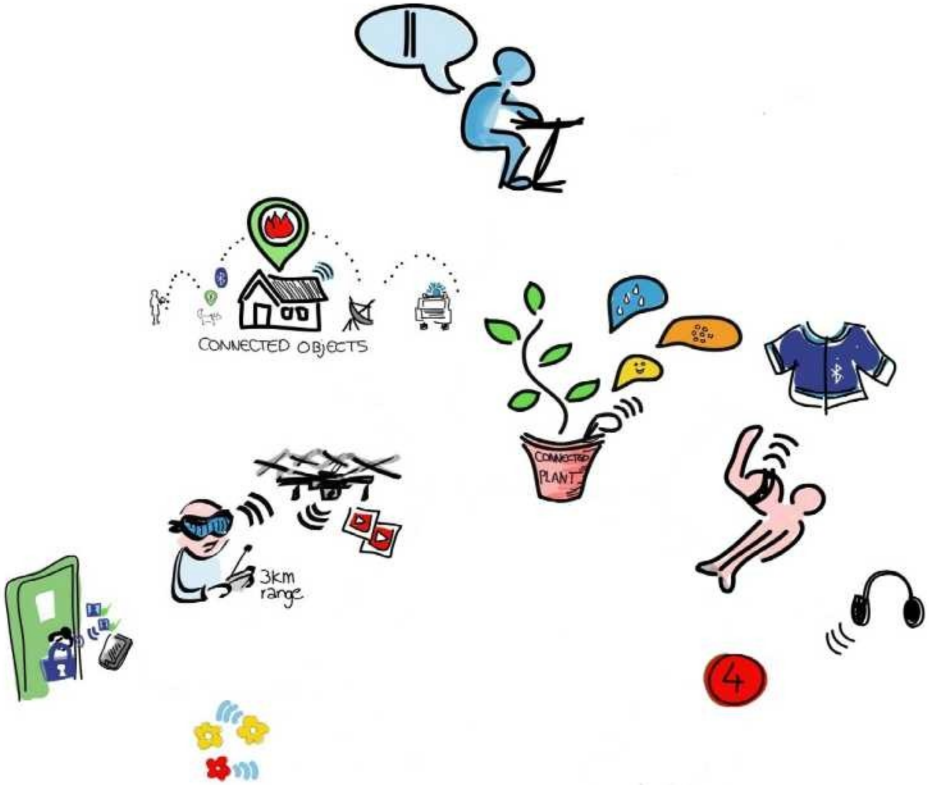
Mifare Classic
MFCUK

Avionics + ADS-B

Hacking MFPs +
PostScript

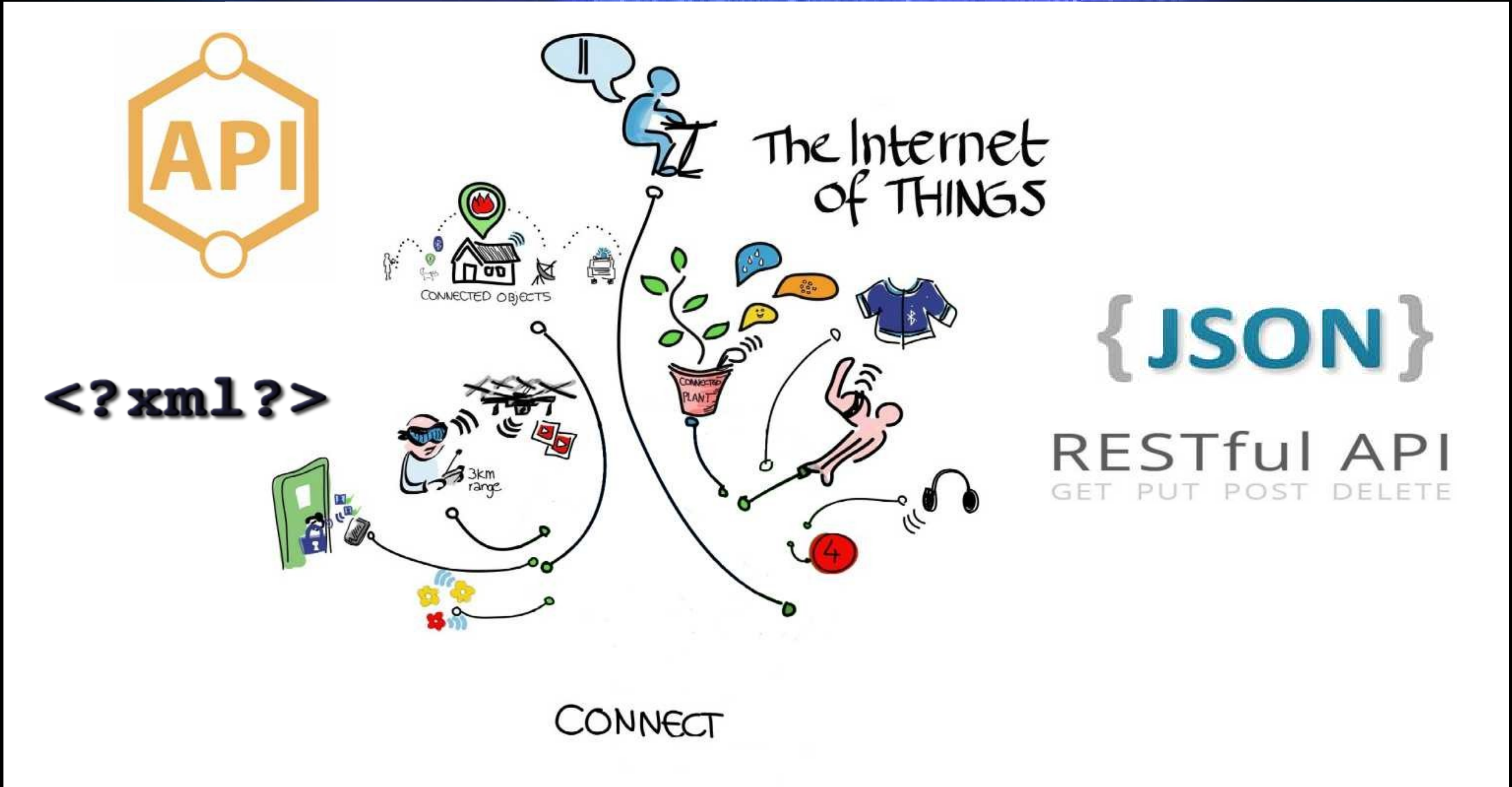
Intro

Embedded Devices Are Everywhere



by Wilgengebroed on Flickr [CC-BY-2.0]

Embedded Devices
Smarter and More Complex



by Wilgengebroed on Flickr [CC-BY-2.0]

Embedded Devices More Interconnected

by Wilgengebroed on Flickr [CC-BY-2.0]

Embedded Software
Firmware is Everywhere

- Embedded devices are diverse – but all of them run software, commonly referred to as firmware

Observations
Magnitude of Embedded/Firmware

- By 2014, there were hundred thousands firmware packages *(Costin et al., USENIX Security 2014)*

Observations
Magnitude of Embedded/Firmware

- By 2014, there were hundred thousands firmware packages *(Costin et al., USENIX Security 14)*
- By 2014, there were 14 billion Internet connected objects *(Cisco, Internet of Things Connections Counter, 2014)*

Observations
Magnitude of Embedded/Firmware

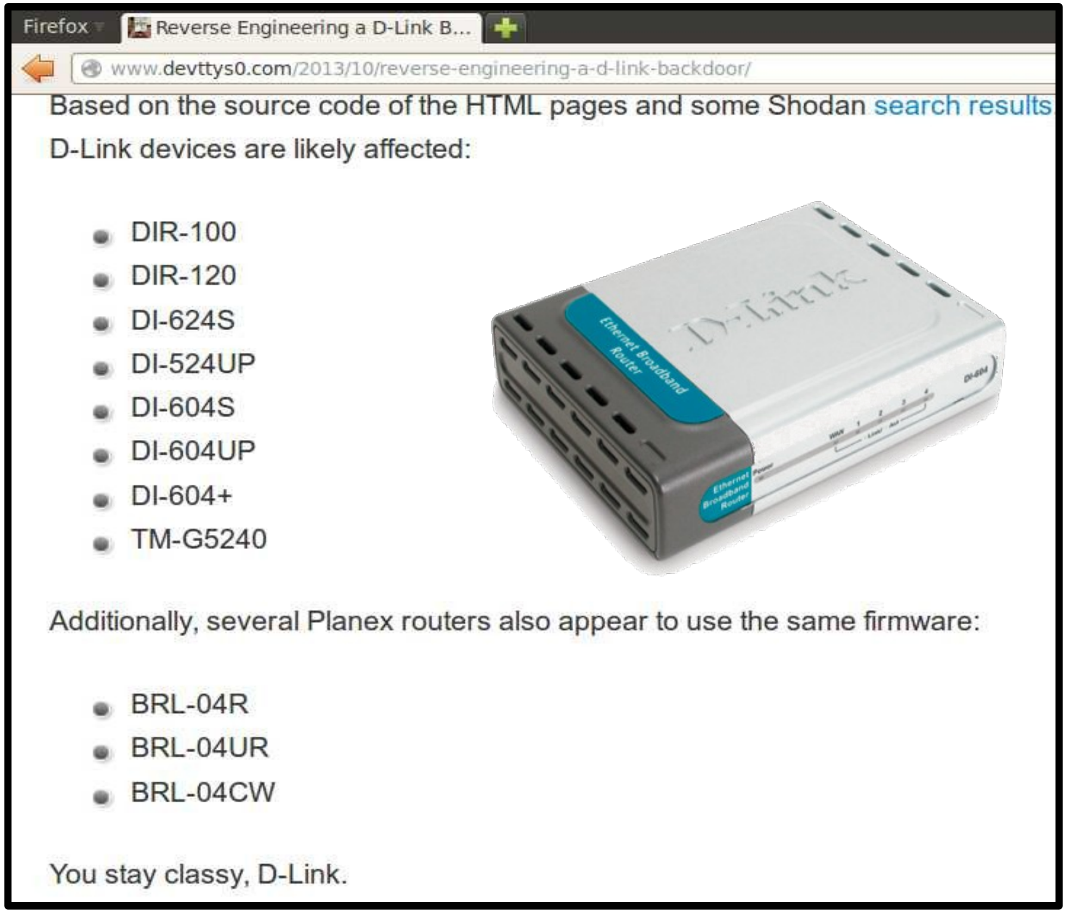
- By 2014, there were hundred thousands firmware packages (*Costin et al., USENIX Security 2014*)
- By 2014, there were 14 billion Internet connected objects (*Cisco, Internet of Things Connections Counter, 2014*)
- By 2020, there will be between 20 and 50 billion interconnected IoT/embedded devices (*Cisco, The Internet of Everything in Motion, 2013*)

Importance of Embedded Systems' Security

- Embedded devices are ubiquitous
  - Even invisible, they are essential to our lives
- Can operate for many years
  - Legacy systems, no (security) updates
- Have a large attack surface
  - Web interfaces
  - Networking services
  - Debug interfaces (forgotten, backdoor)
  - ...

Many Examples of Insecure Embedded Systems

- Routers


Firefox
Reverse Engineering a D-Link B...
www.devttys0.com/2013/10/reverse-engineering-a-d-link-backdoor/
D-Link
Ethernet Broadband Router

Based on the source code of the HTML pages and some Shodan search results
D-Link devices are likely affected:

- DIR-100
- DIR-120
- DI-624S
- DI-524UP
- DI-604S
- DI-604UP
- DI-604+
- TM-G5240

Additionally, several Planex routers also appear to use the same firmware:

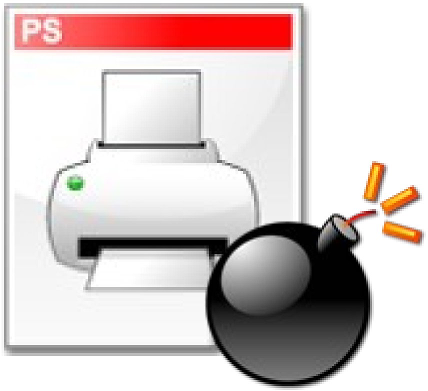
- BRL-04R
- BRL-04UR
- BRL-04CW

You stay classy, D-Link.

Many Examples of Insecure Embedded Systems

- Routers
- Printers

Networked printers at risk
(30/12/2011, McAfee Labs)



Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP

Cisco VoIP Phones Affected By On Hook Security Vulnerability (12/06/2012, Forbes)

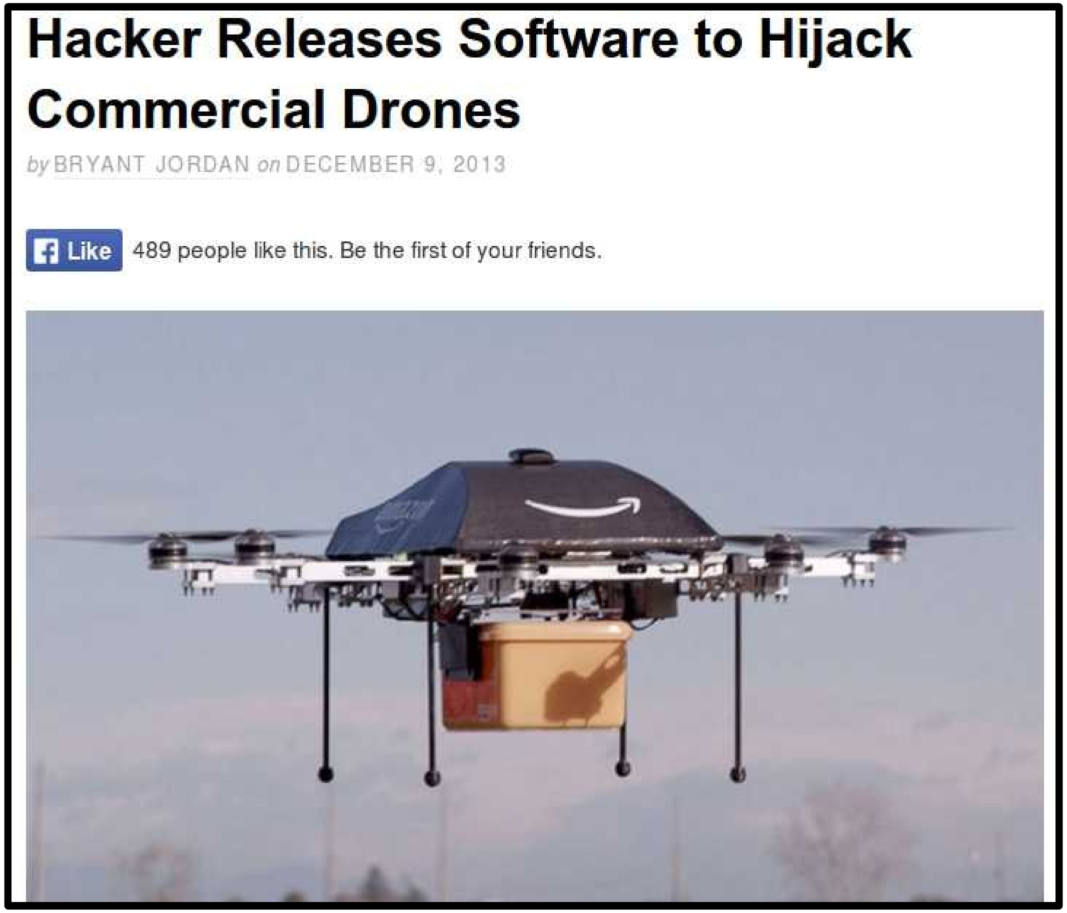
Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars

Hackers Reveal Nasty New Car Attacks – With Me Behind The Wheel (12/08/2013, Forbes)

Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones



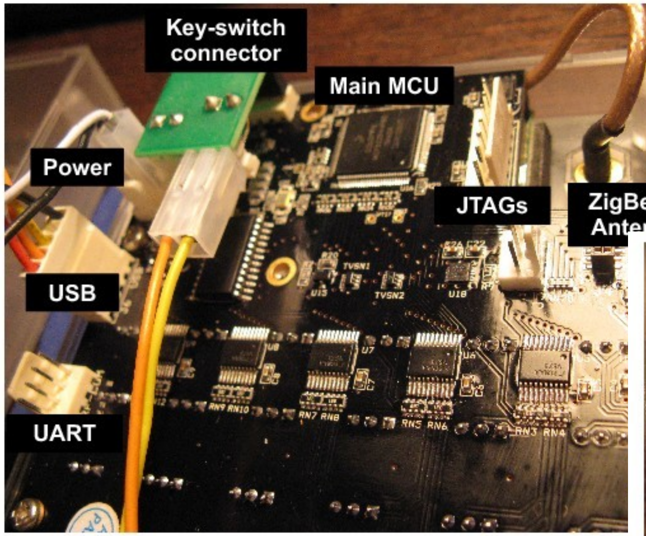
Hacker Releases Software to Hijack Commercial Drones

by BRYANT JORDAN *on* DECEMBER 9, 2013

Like 489 people like this. Be the first of your friends.

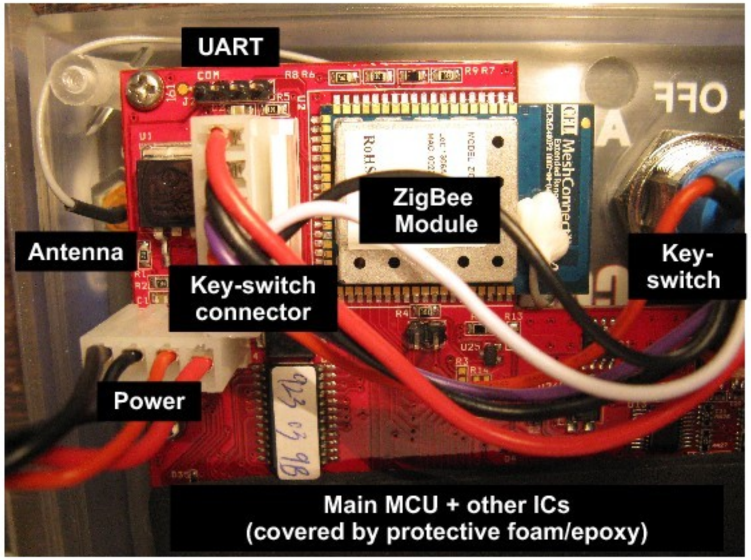
Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones
- Fireworks



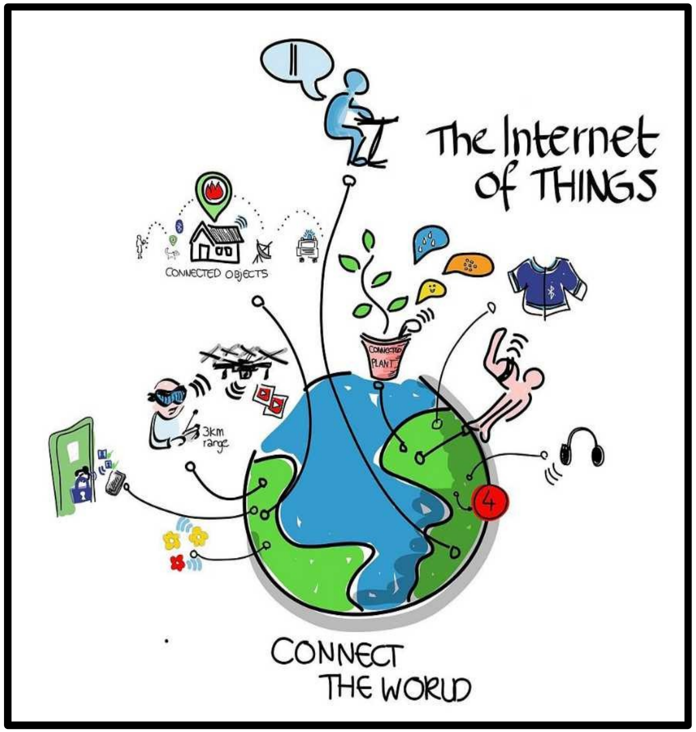
Firing Module

Remote Control



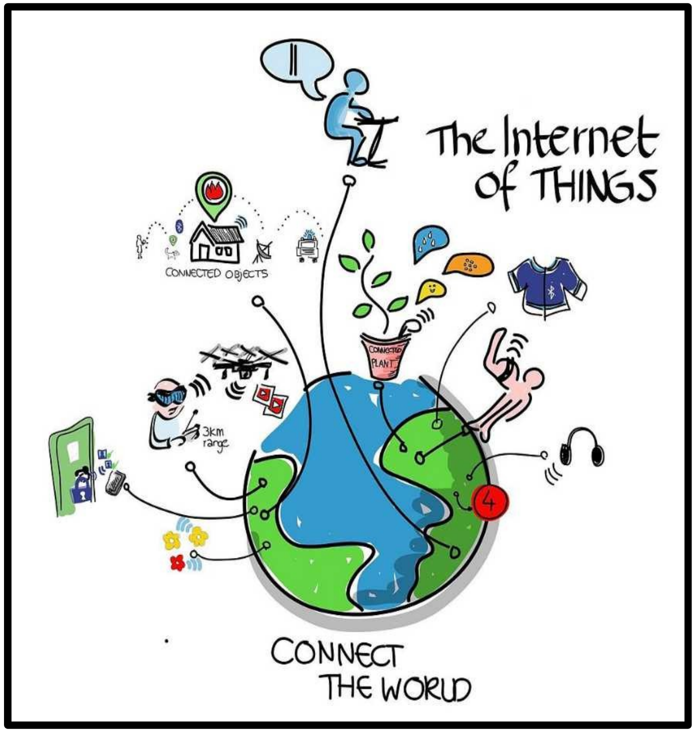
Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones
- Fireworks
- Etc.



Many Examples of Insecure Embedded Systems

- Routers
- Printers
- VoIP
- Cars
- Drones
- Fireworks
- Etc.



Each of the above is a result of individual analysis

Manual and tedious efforts → Does not scale

Review
Manual Analysis Process

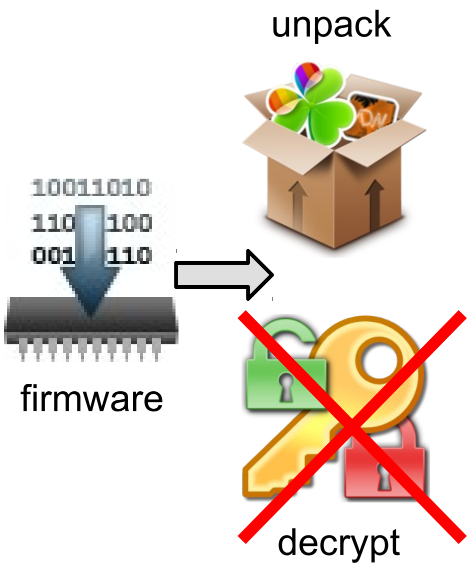
-

firmware

firmware · ǝɹ

Review
Manual Analysis Process

-



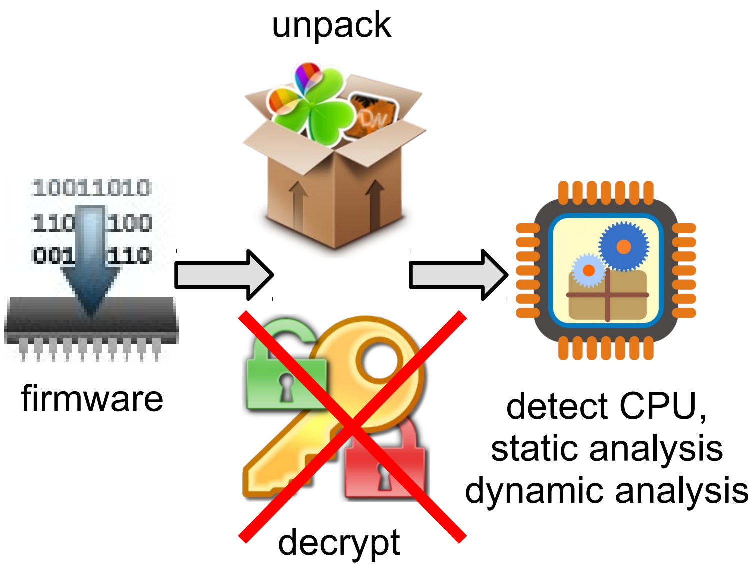
IHEX format

```
:100000000C942A000C9434000C9434000C943400AA
:100010000C9434000C9434000C9434000C94340090
:100020000C9434000C9434000C9434000C94340080
:100030000C9434000C9434000C9434000C94340070
:100040000C9434000C9434000C9434000C94340060
:100050000C94340011241FBECFE5D8E0DEBFCDBF25
:100060000E9436000C9445000C9400008FEF87BB73
:100070002CE231E088B3809588BB80E197E2F901FA
:0E0080003197F1F70197D9F7F5CFF894FFCF3C
:00000001FF
```

plain text firmware

Review
Manual Analysis Process

-



unpack

firmware

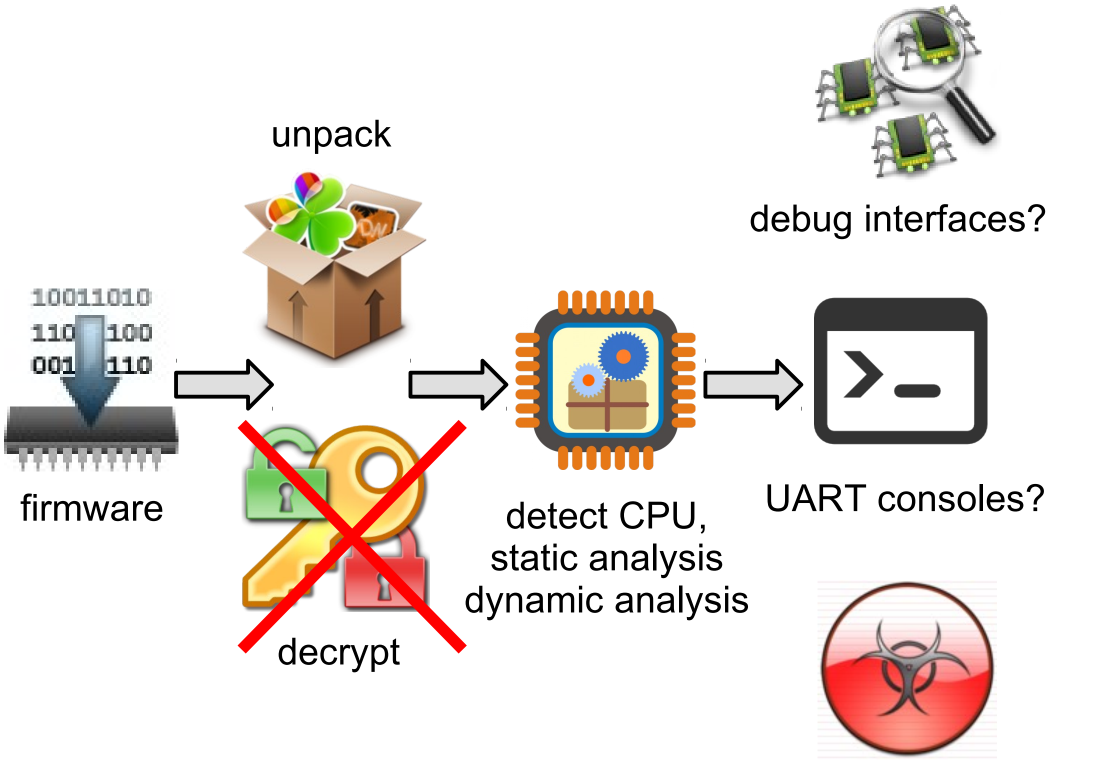
decrypt

detect CPU,
static analysis
dynamic analysis

Motorola m68k-based CPU

Review
Manual Analysis Process

-



firmware

unpack

~~decrypt~~

detect CPU, static analysis dynamic analysis

debug interfaces?

UART consoles?

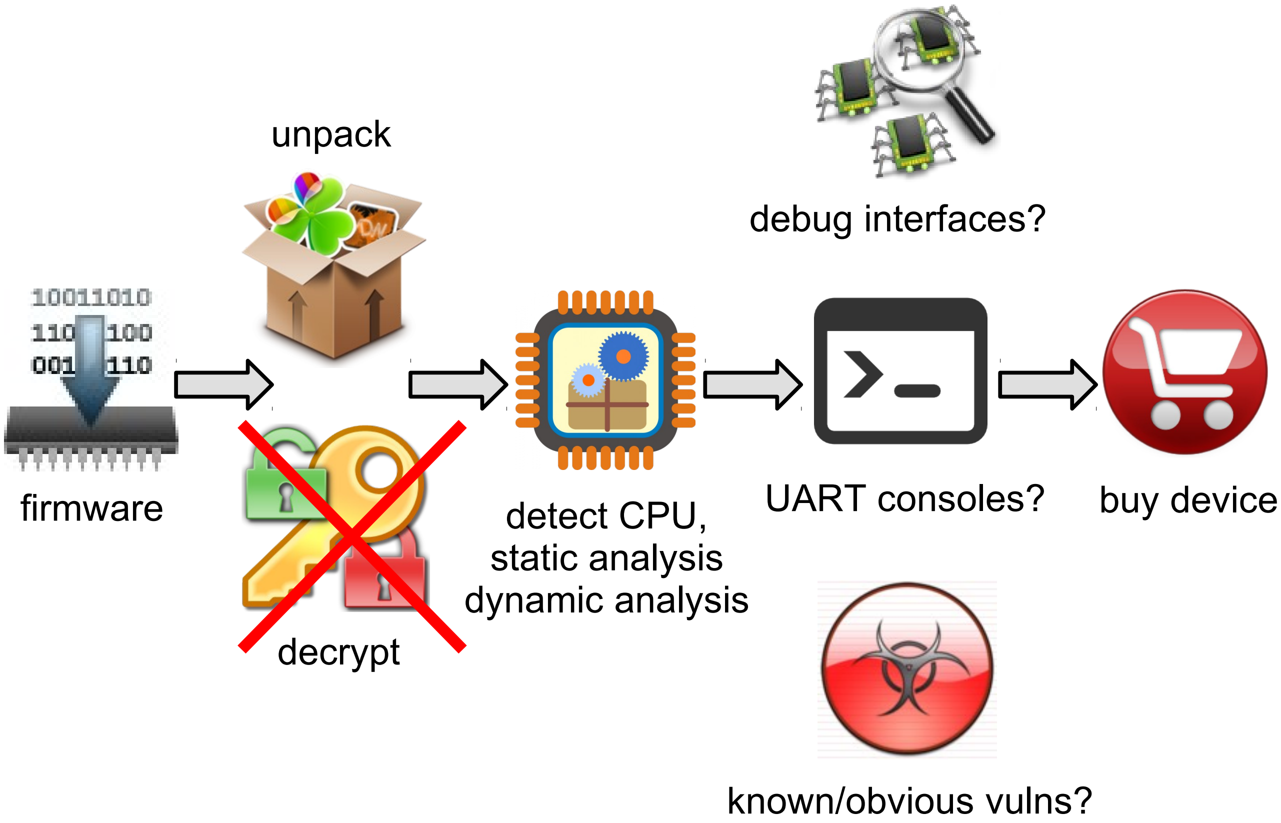
known/obvious vulns?

UART “boot>” prompts

802.15.4 functions

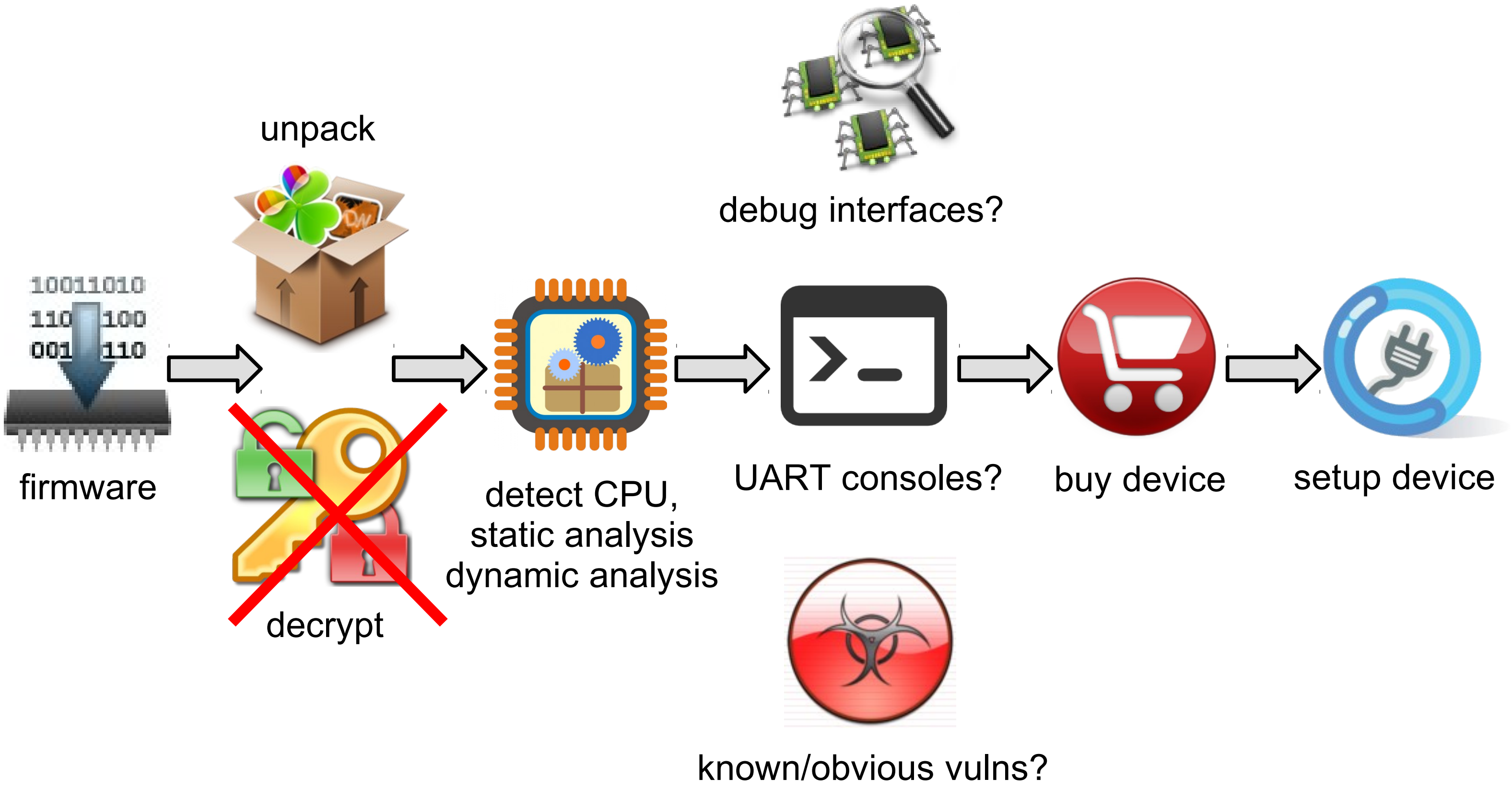
Review
Manual Analysis Process

-



Review
Manual Analysis Process

-

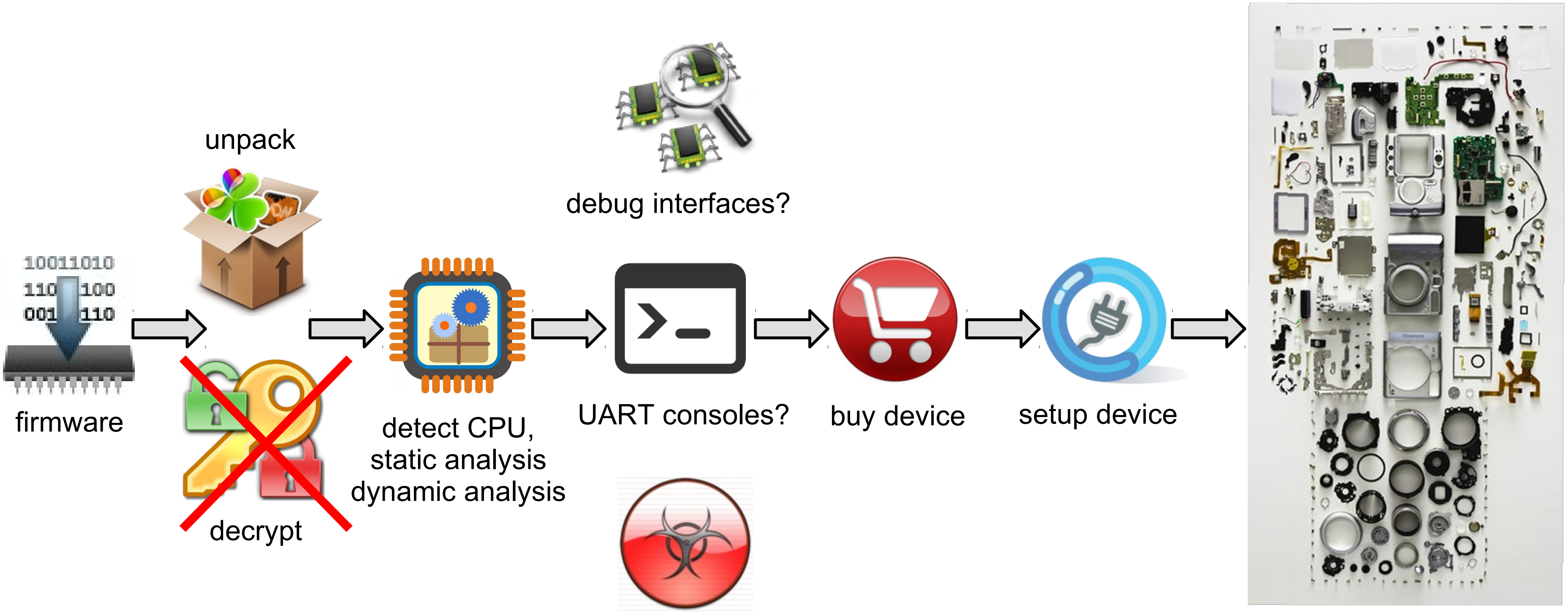


firmware → unpack / ~~decrypt~~ → detect CPU, static analysis dynamic analysis → UART consoles? → buy device → setup device

debug interfaces?

known/obvious vulns?

Review
Manual Analysis Process



firmware

unpack

~~decrypt~~

detect CPU, static analysis dynamic analysis

debug interfaces?

UART consoles?

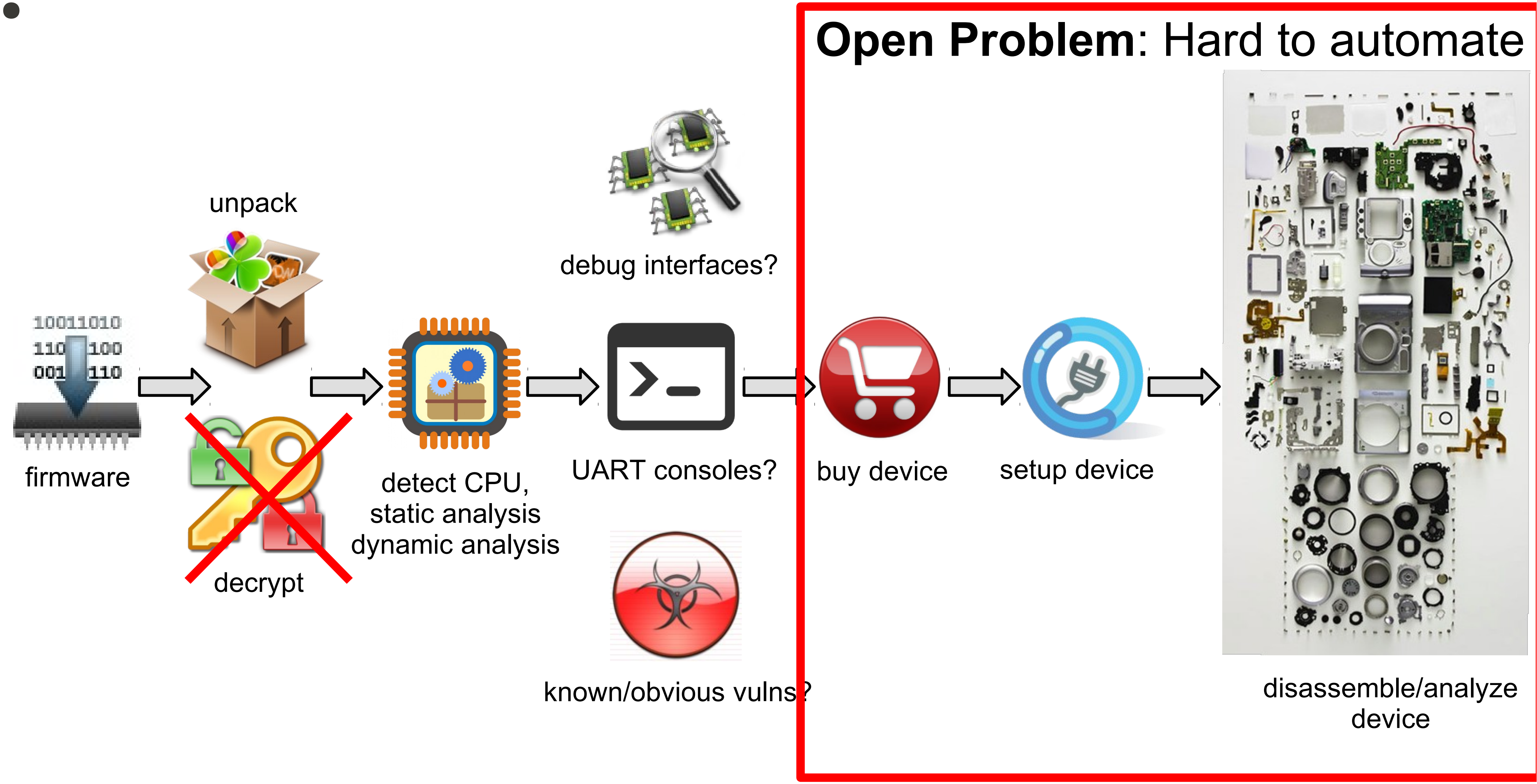
known/obvious vulns?

buy device

setup device

disassemble/analyze device

Review
Manual Analysis Process



Open Problem: Hard to automate

firmware

unpack

decrypt

detect CPU, static analysis dynamic analysis

debug interfaces?

UART consoles?

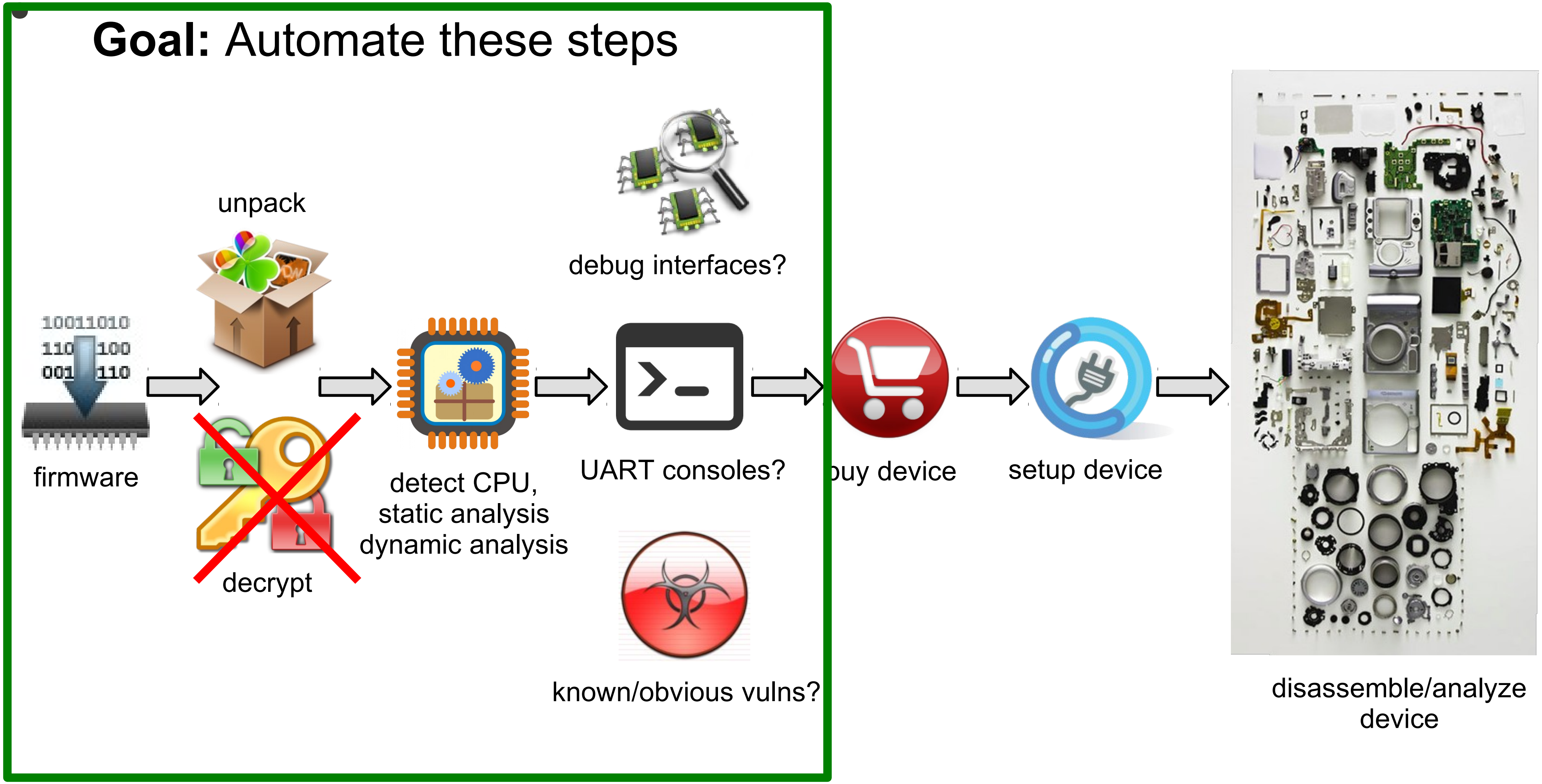
known/obvious vulns?

buy device

setup device

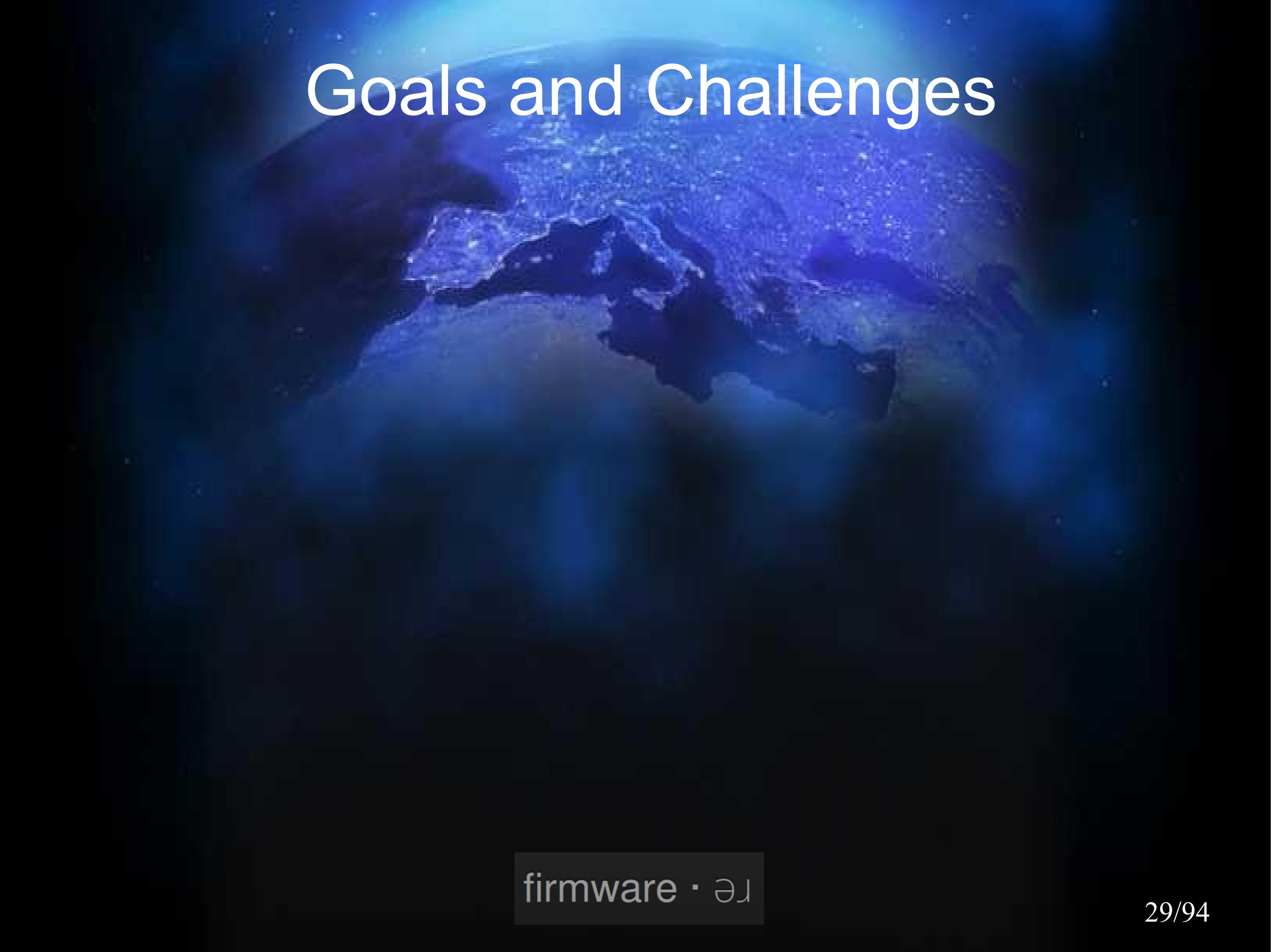
disassemble/analyze device

Review
Manual Analysis Process



Goal: Automate these steps

Goals and Challenges



Idea → Goal

Perform large scale automated analysis to better understand, classify and analyze firmware images, without using devices

Challenges

- Large number of devices
- Large number of firmware files
- Highly heterogeneous systems
- Increasingly “smart”, “connected”
- Highly unstructured firmware data
- Vulnerable devices exposed

Challenges → Solutions

- Large number of devices → Analysis without devices
- Large number of firmware files → Scalable architectures
- Highly heterogeneous systems → Generic techniques
- Increasingly “smart”, “connected” → Focus on web interfaces & APIs
- Highly unstructured firmware data → Large dataset classification
- Vulnerable devices exposed → Technology-independent device fingerprinting

Large Scale Challenge 1: Firmware and Device Classification

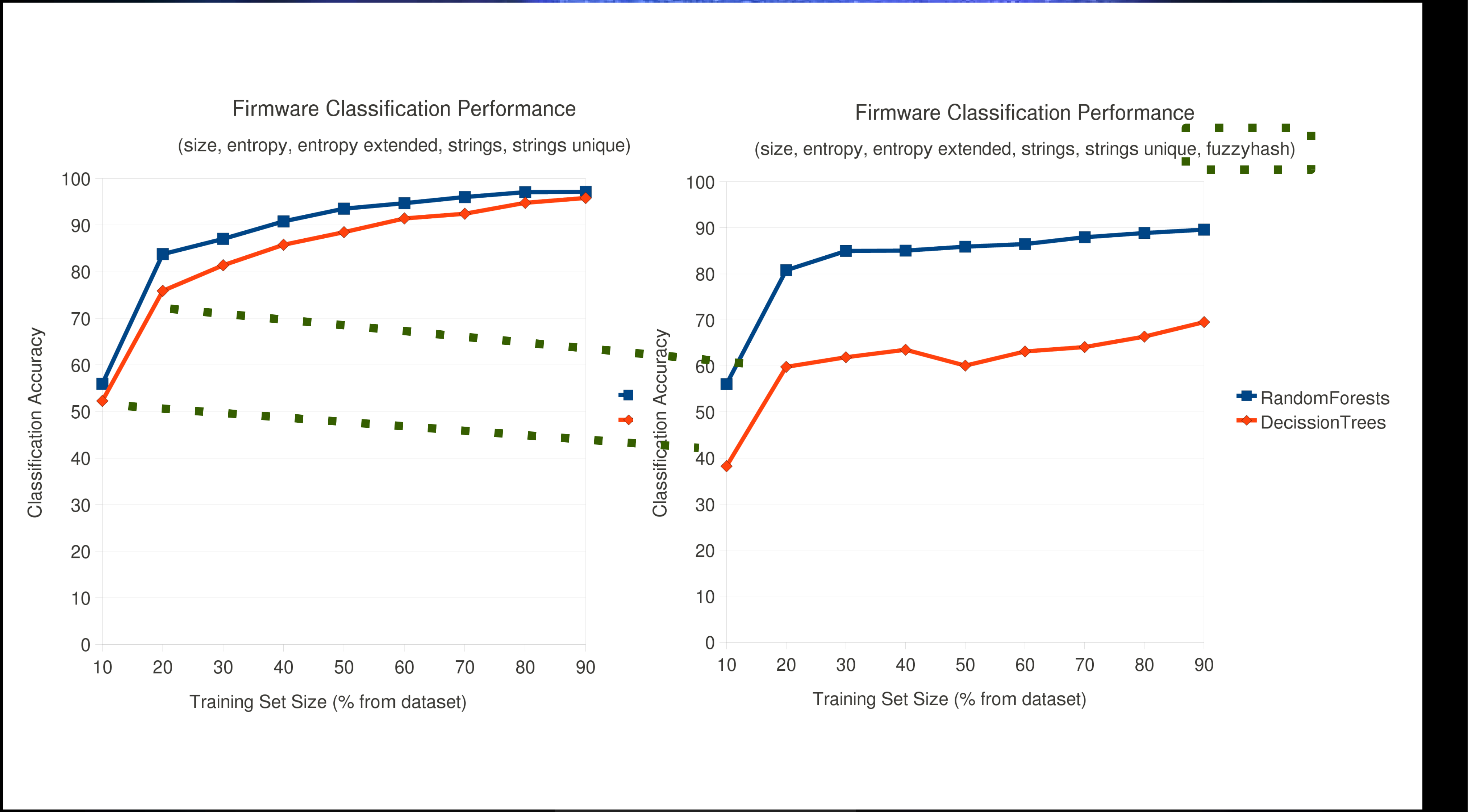
Firmware Classification Why and How?

- Why?
  - There are hundred thousands firmware packages (*Costin et al., USENIX Security 2014*)
  - Any volunteer for manual triage? :)
- How?
  - Machine Learning (ML)
  - E.g., python's scikit-learn

Firmware Classification ML Details

- Random Forests, Decision Trees
- File size
- Entropy value
- Extended entropy information
- Category strings
- Category unique strings

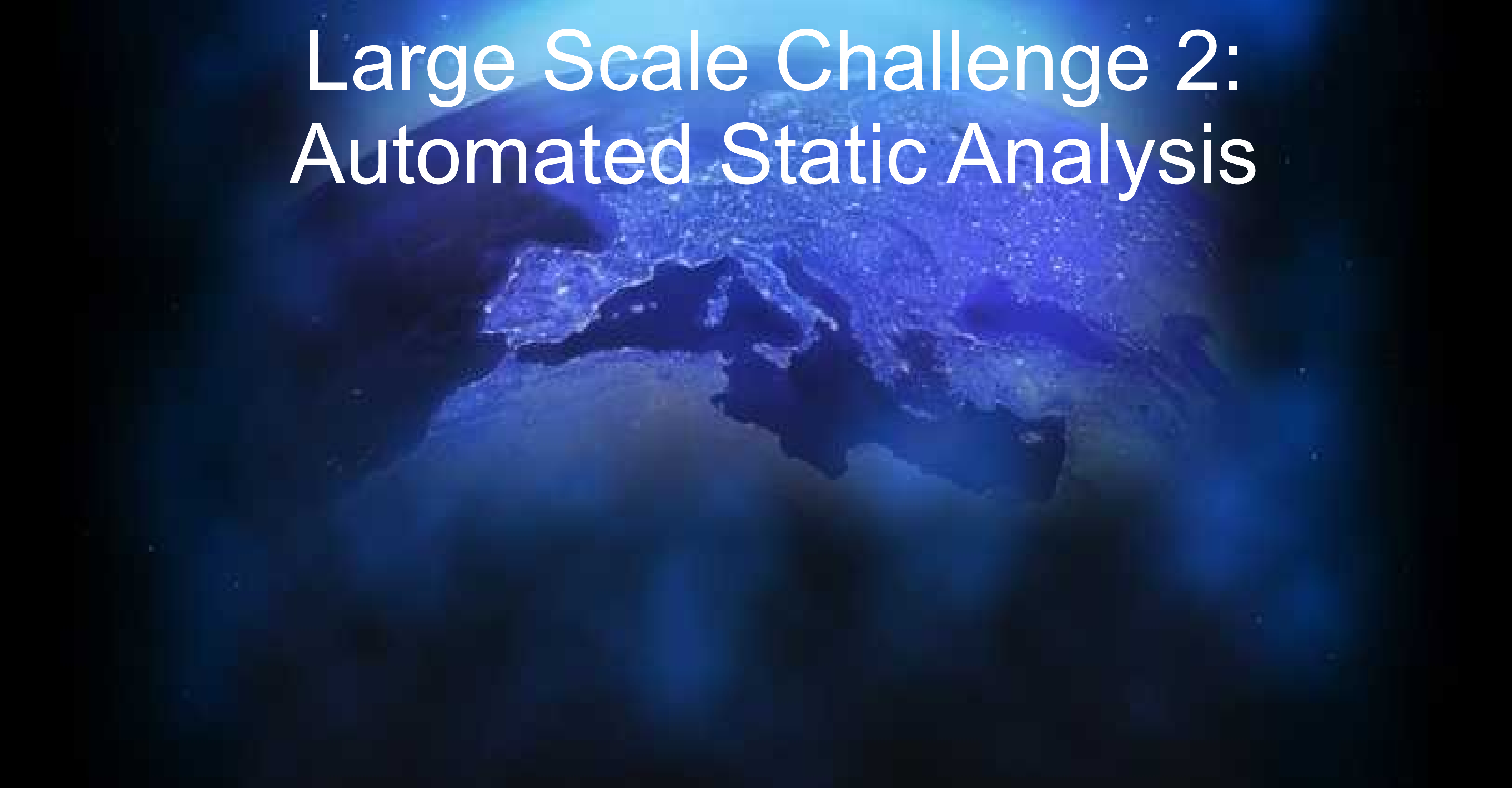
Firmware Classification ML Examples



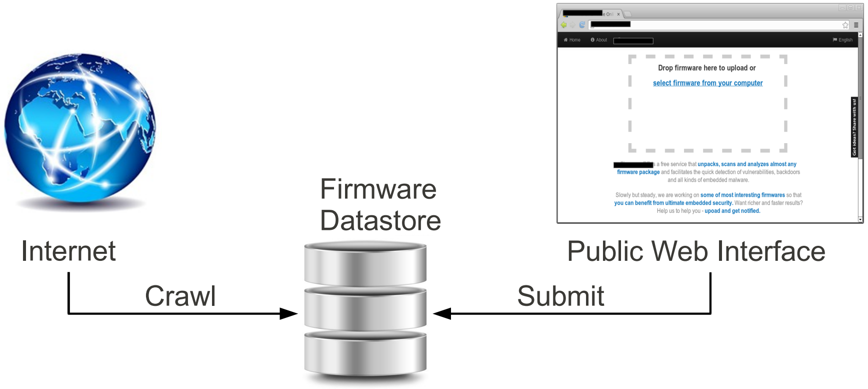
Firmware Classification ML Summary

- The local optimum for our setup
  - Features [*size, entropy, entropy extended, category strings, category unique strings*]
  - Random Forests classifier
  - Training sets based on 40% of each category
  - Achieves more than 90% accuracy

Large Scale Challenge 2: Automated Static Analysis



Static Firmware Analysis Automated and Large Scale



[redacted] e Onli ×

[redacted]

Home About [redacted] English

Drop firmware here to upload or

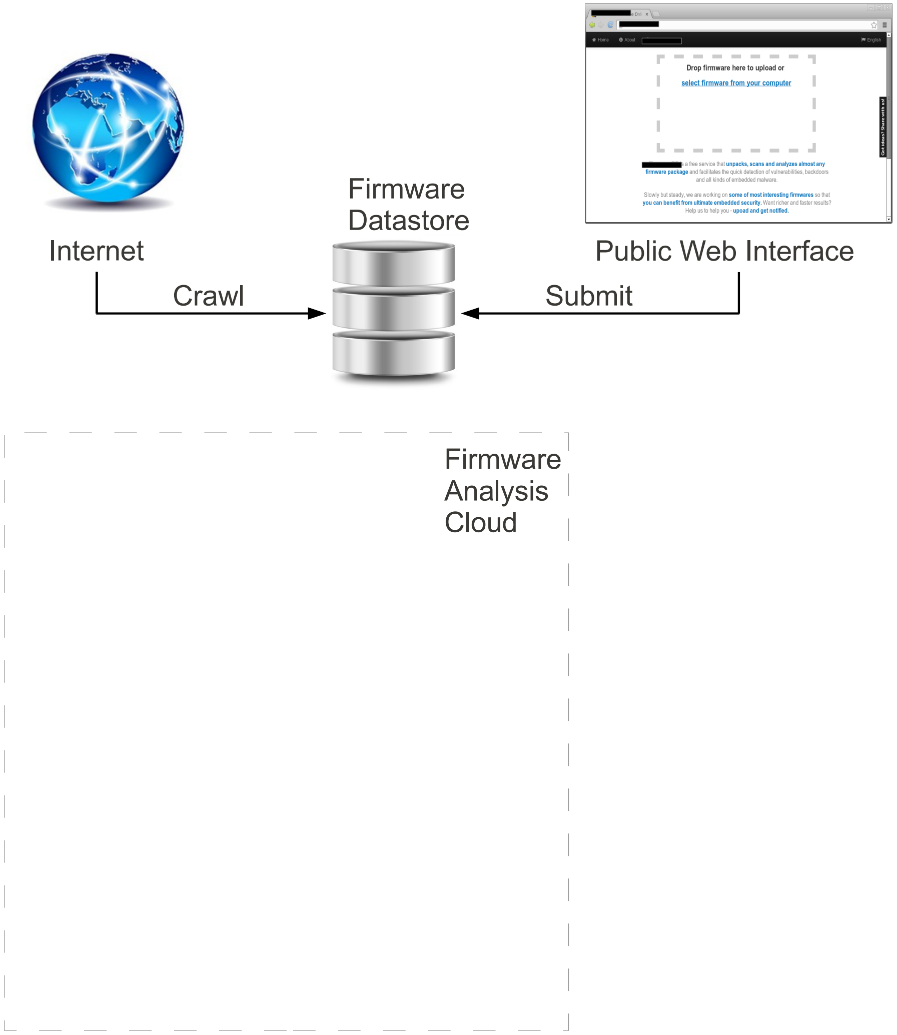
select firmware from your computer

Got ideas? Share with us!

[redacted] s a free service that unpacks, scans and analyzes almost any firmware package and facilitates the quick detection of vulnerabilities, backdoors and all kinds of embedded malware.

Slowly but steady, we are working on some of most interesting firmwares so that you can benefit from ultimate embedded security. Want richer and faster results? Help us to help you - upoad and get notified.

Static Firmware Analysis Automated and Large Scale



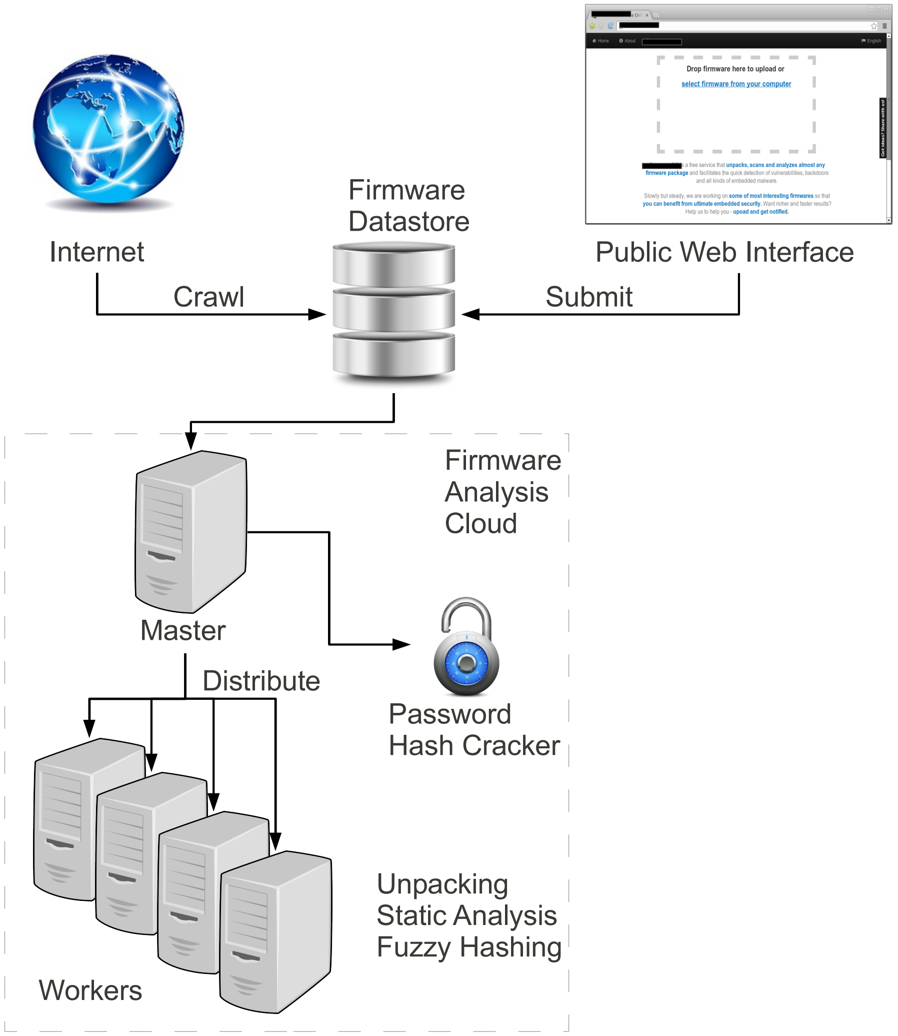
Drop firmware here to upload or

select firmware from your computer

[redacted]s a free service that **unpacks, scans and analyzes almost any firmware package** and facilitates the quick detection of vulnerabilities, backdoors and all kinds of embedded malware.

Slowly but steady, we are working on **some of most interesting firmwares** so that **you can benefit from ultimate embedded security.** Want richer and faster results? Help us to help you - **upoad and get notified.**

Static Firmware Analysis Automated and Large Scale



[redacted]

[redacted]

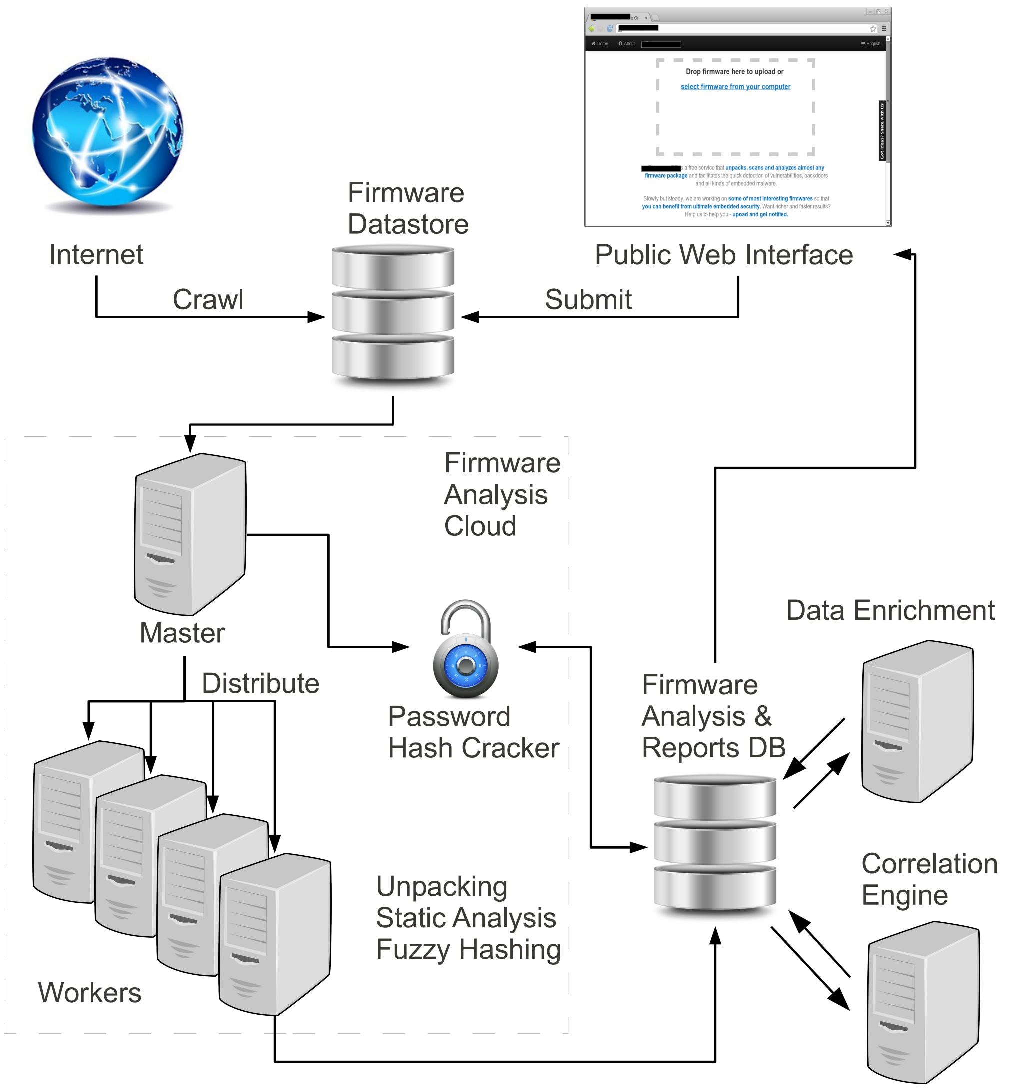
Drop firmware here to upload or

select firmware from your computer

[redacted] a free service that unpacks, scans and analyzes almost any firmware package and facilitates the quick detection of vulnerabilities, backdoors and all kinds of embedded malware.

Slowly but steady, we are working on some of most interesting firmwares so that you can benefit from ultimate embedded security. Want richer and faster results? Help us to help you - upoad and get notified.

Static Firmware Analysis
Automated and Large Scale



Drop firmware here to upload or

select firmware from your computer

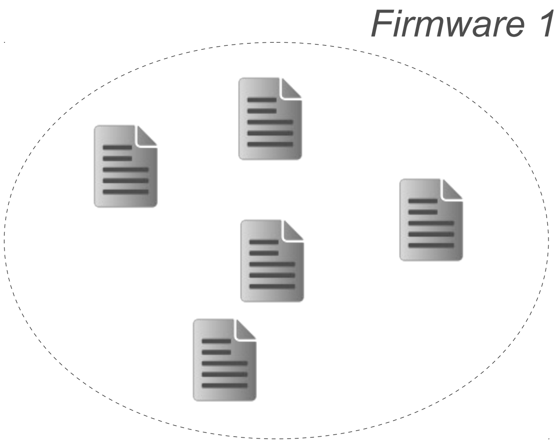
[redacted] s a free service that unpacks, scans and analyzes almost any firmware package and facilitates the quick detection of vulnerabilities, backdoors and all kinds of embedded malware.

Slowly but steady, we are working on some of most interesting firmwares so that you can benefit from ultimate embedded security. Want richer and faster results? Help us to help you - upoad and get notified.

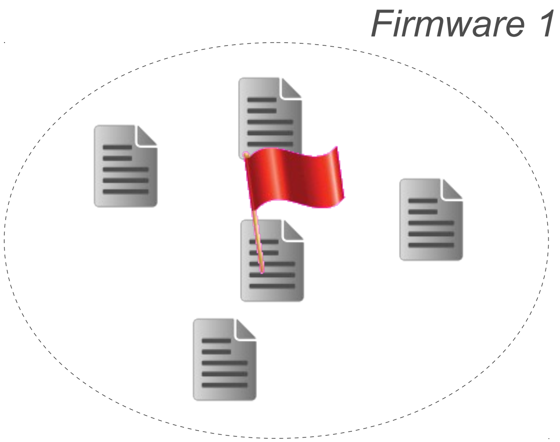
Static Firmware Analysis
Types of Tests

- Misconfiguration
  - Web-server configs, Code repositories
- Credentials
  - Weak/Default/Hard-coded
- Data enrichment
  - Versions → Software packages
  - Keywords → Known problems (telnet, shell, UART, backdoor)
- Correlation and clustering
  - Based on: Fuzzy hashes, Private SSL keys, Credentials

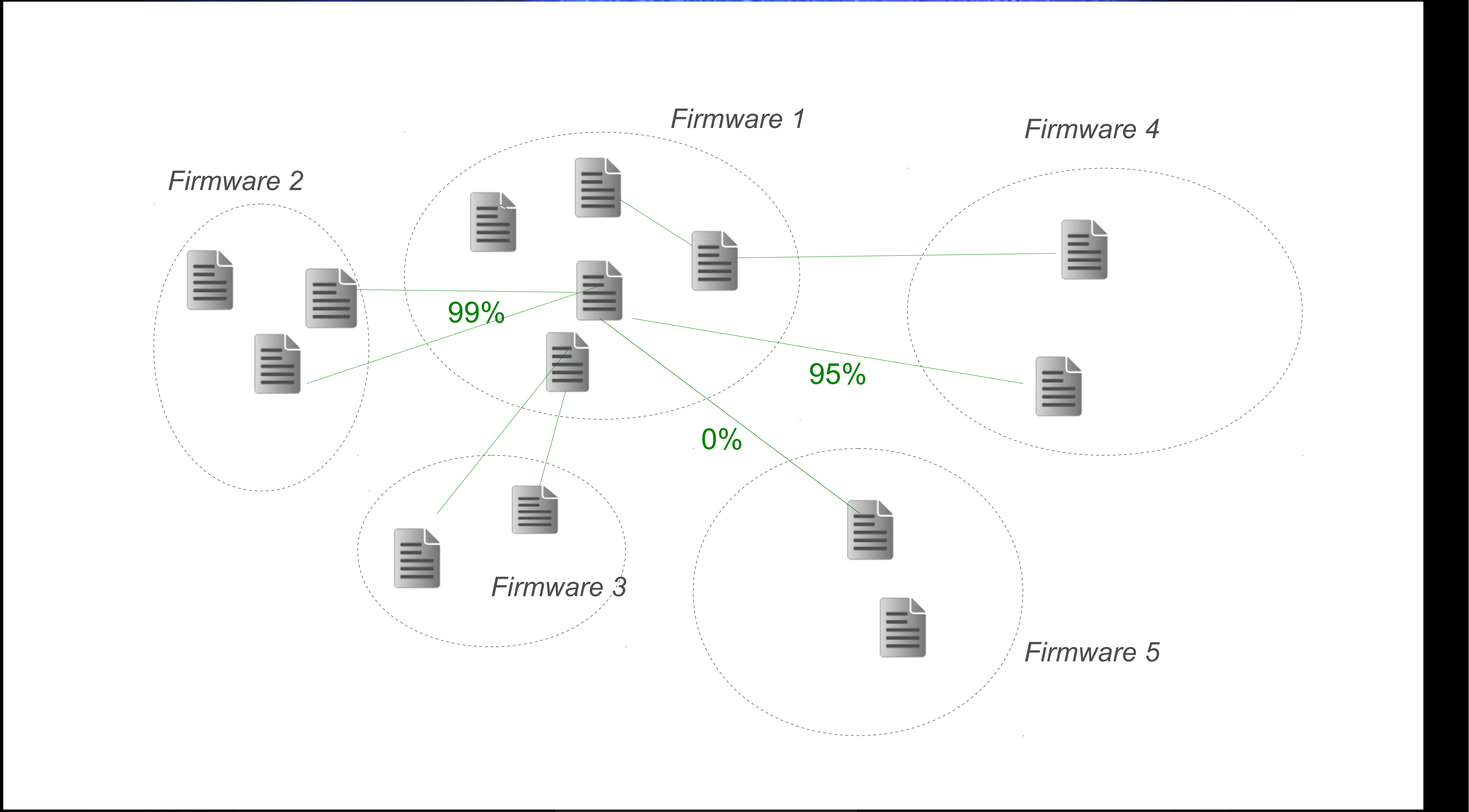
Example:
Firmware content correlation



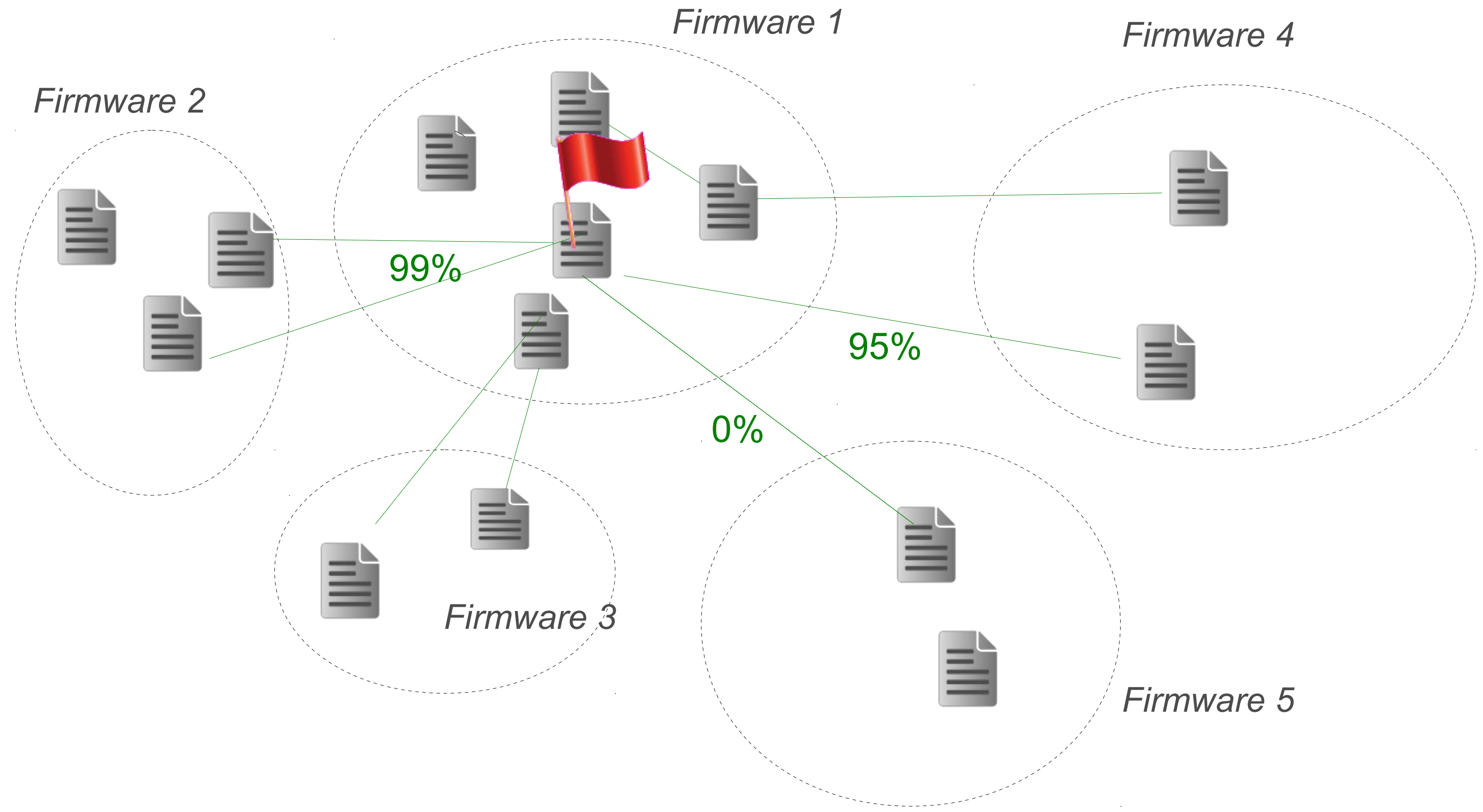
Example:
Firmware content correlation



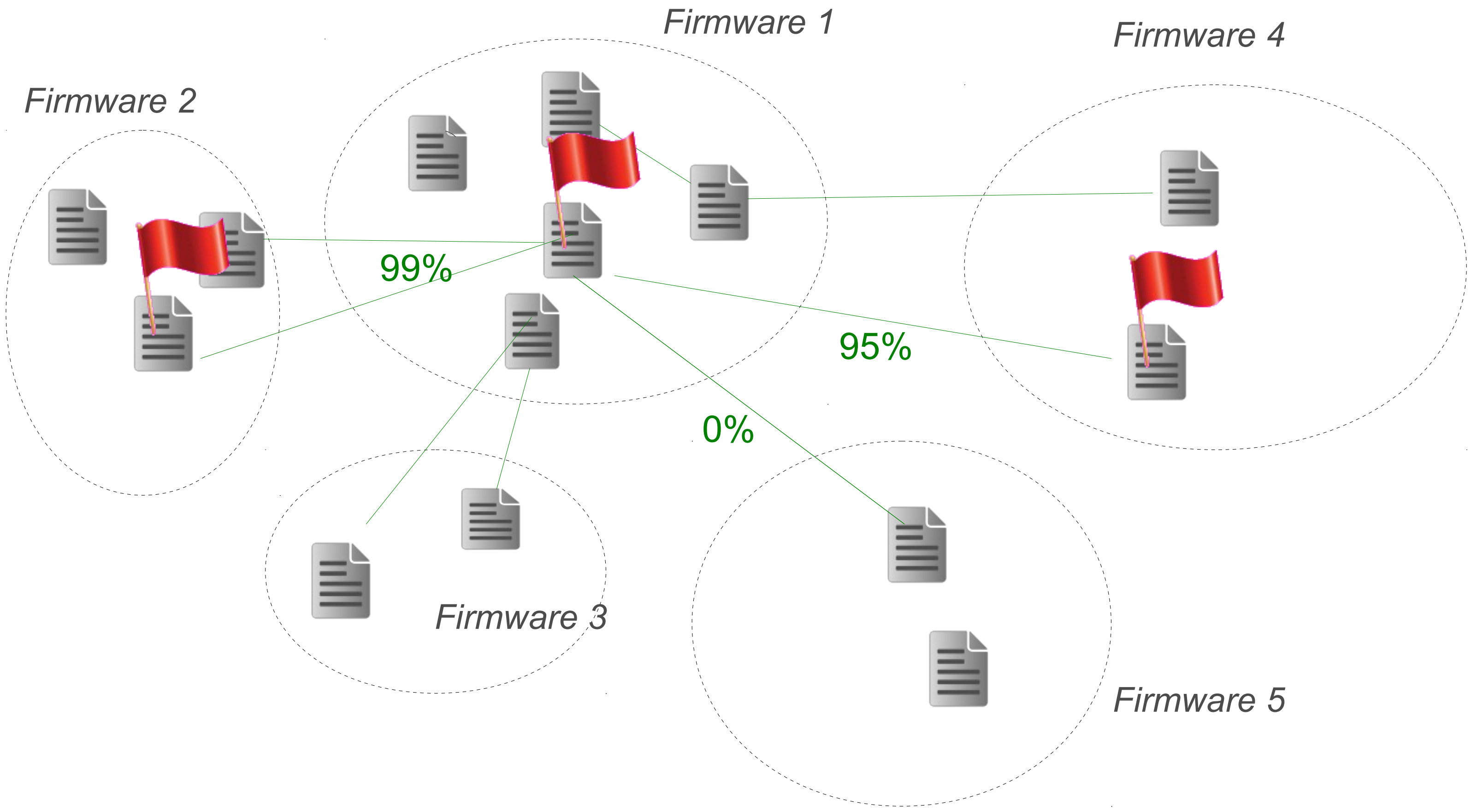
Example:
Firmware content correlation



Example:
Firmware content correlation

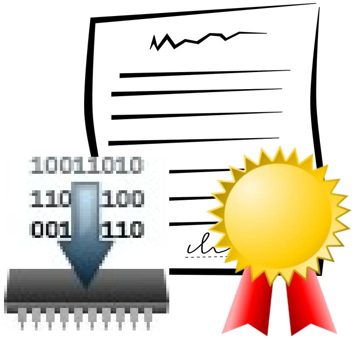


Example:
Firmware content correlation

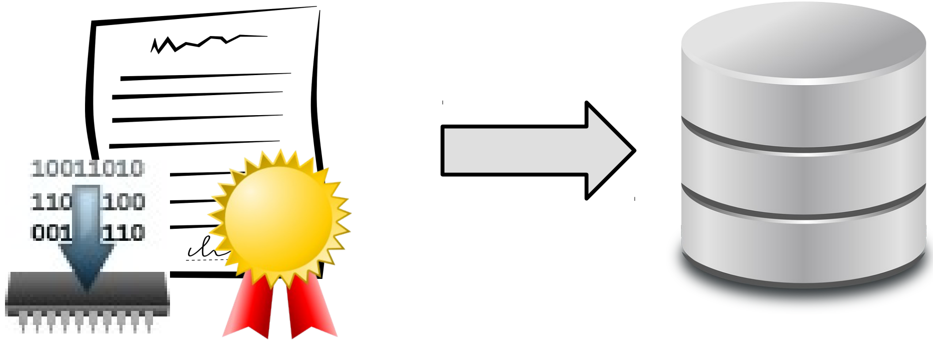


Example: Firmware HTTPS keys correlation

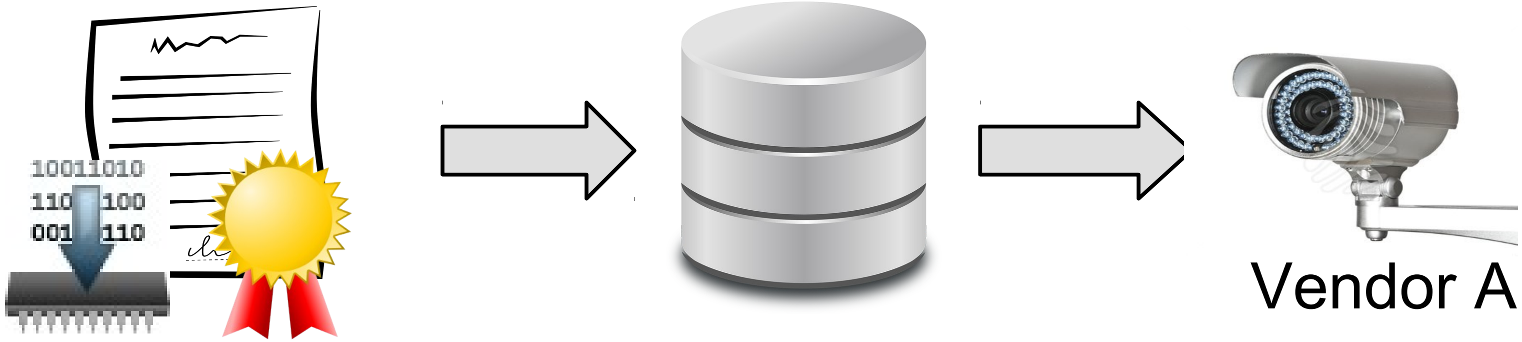
Example:
Firmware HTTPS keys correlation



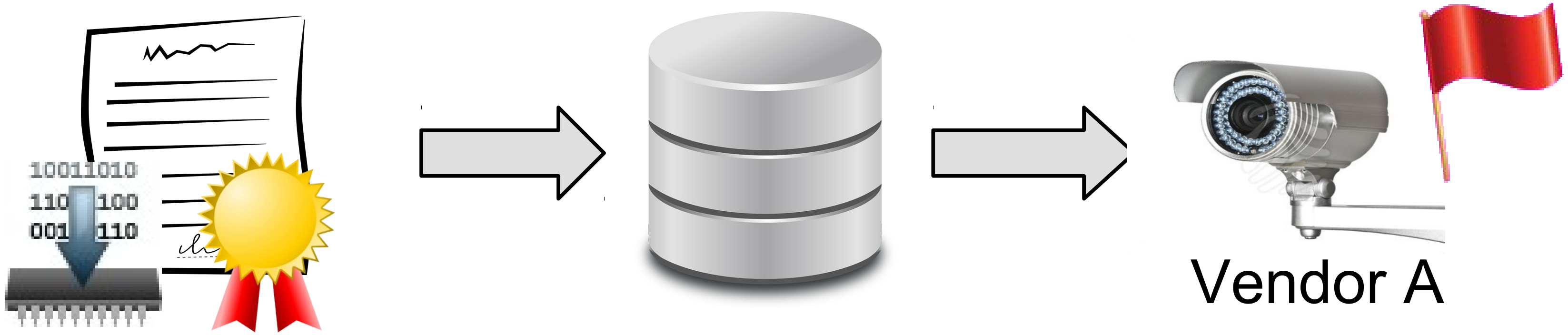
Example:
Firmware HTTPS keys correlation



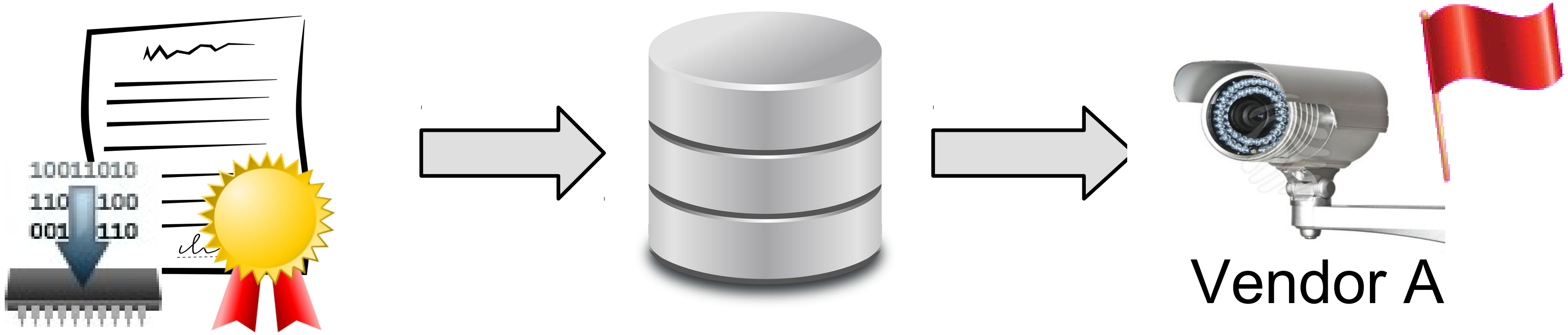
Example:
Firmware HTTPS keys correlation



Example: Firmware HTTPS keys correlation

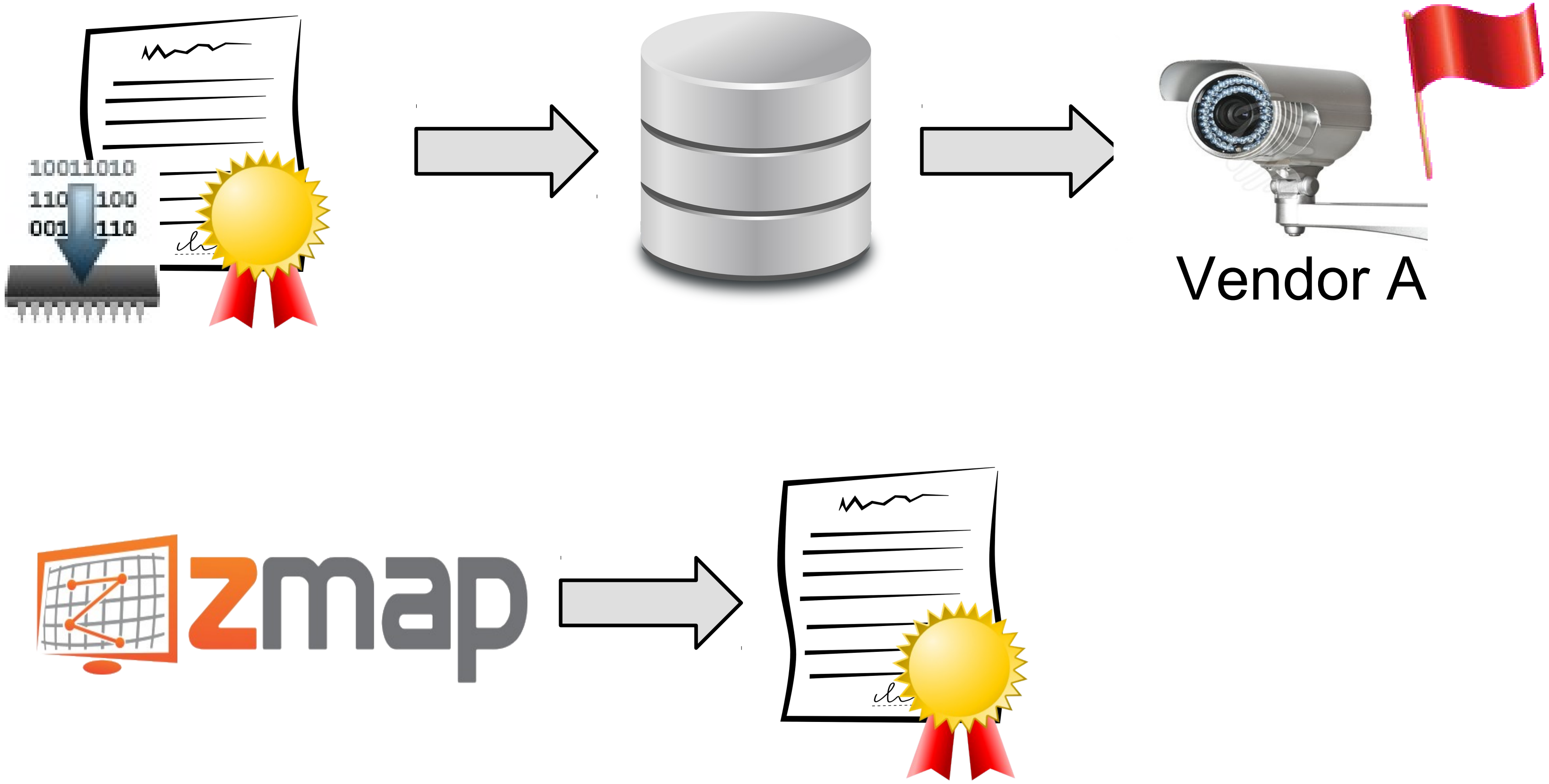


Example:
Firmware HTTPS keys correlation

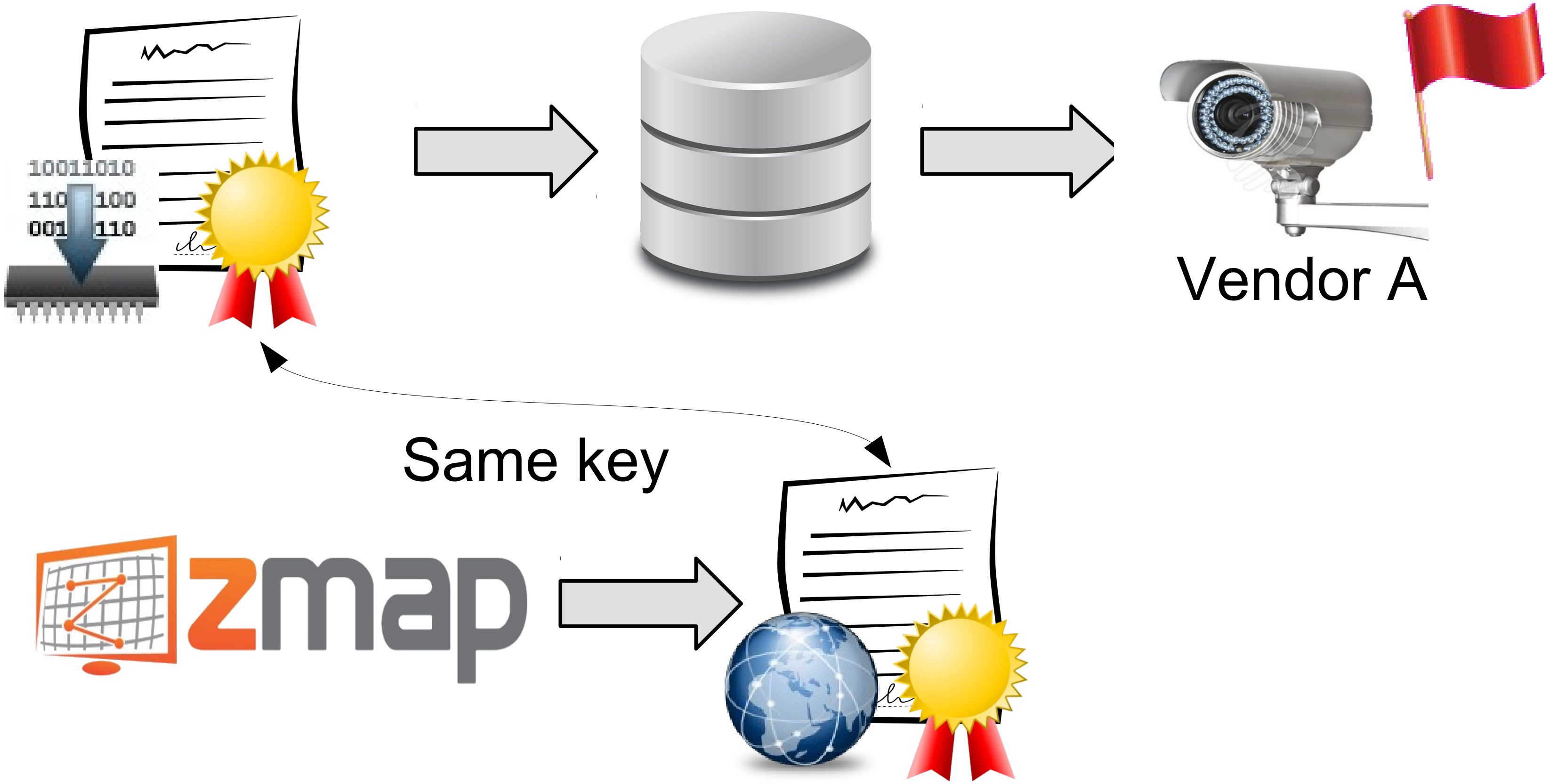


zmap

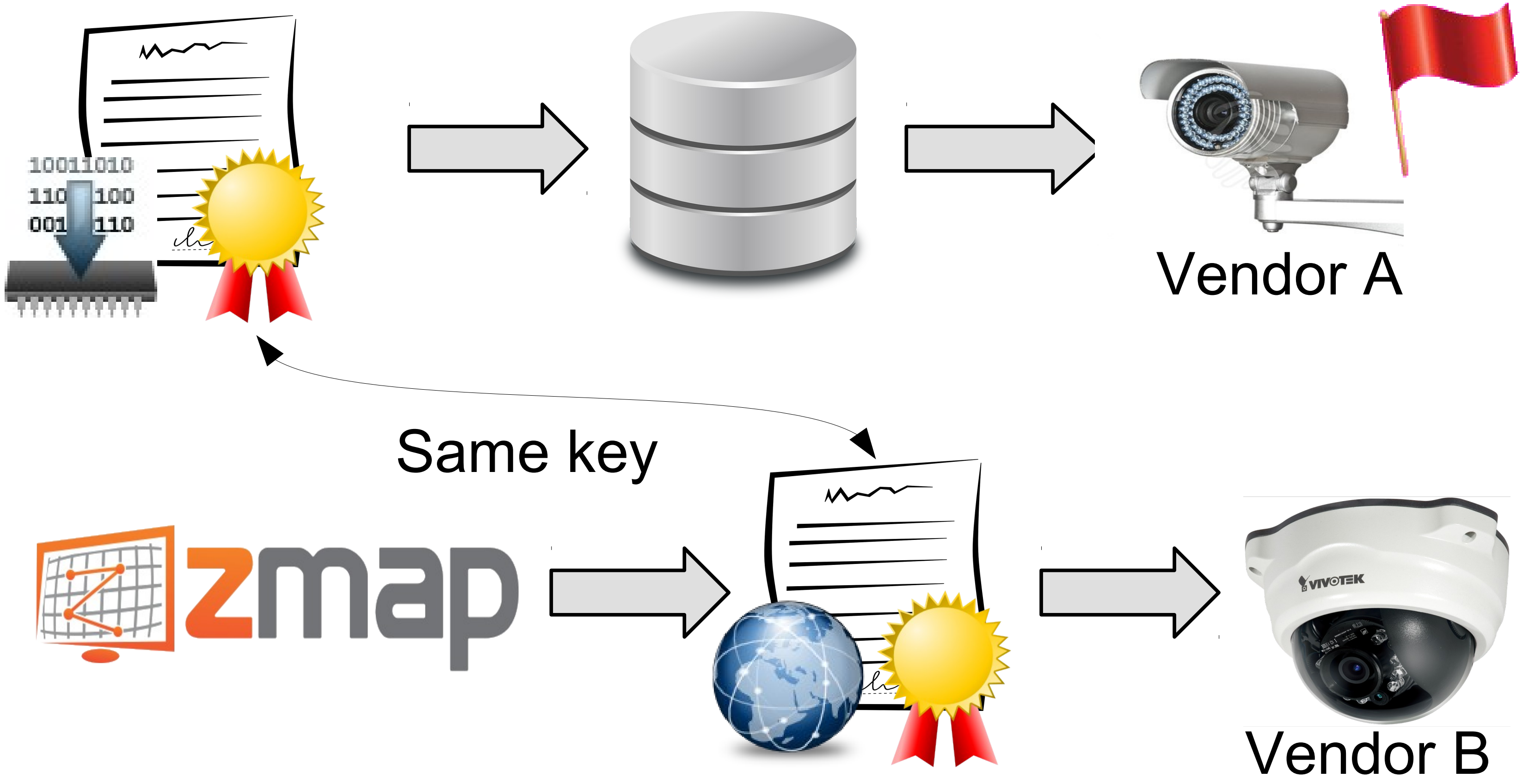
Example:
Firmware HTTPS keys correlation



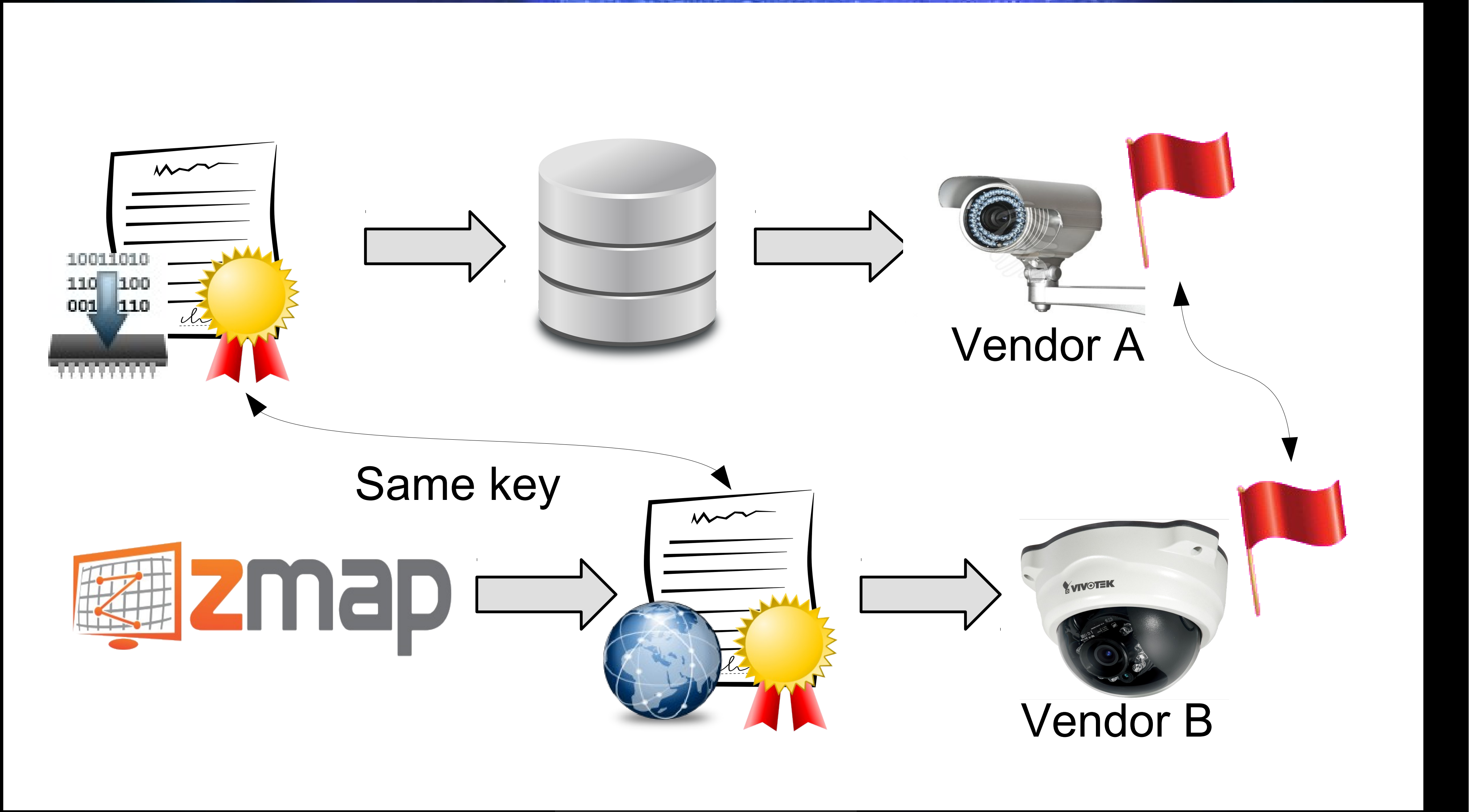
Example:
Firmware HTTPS keys correlation



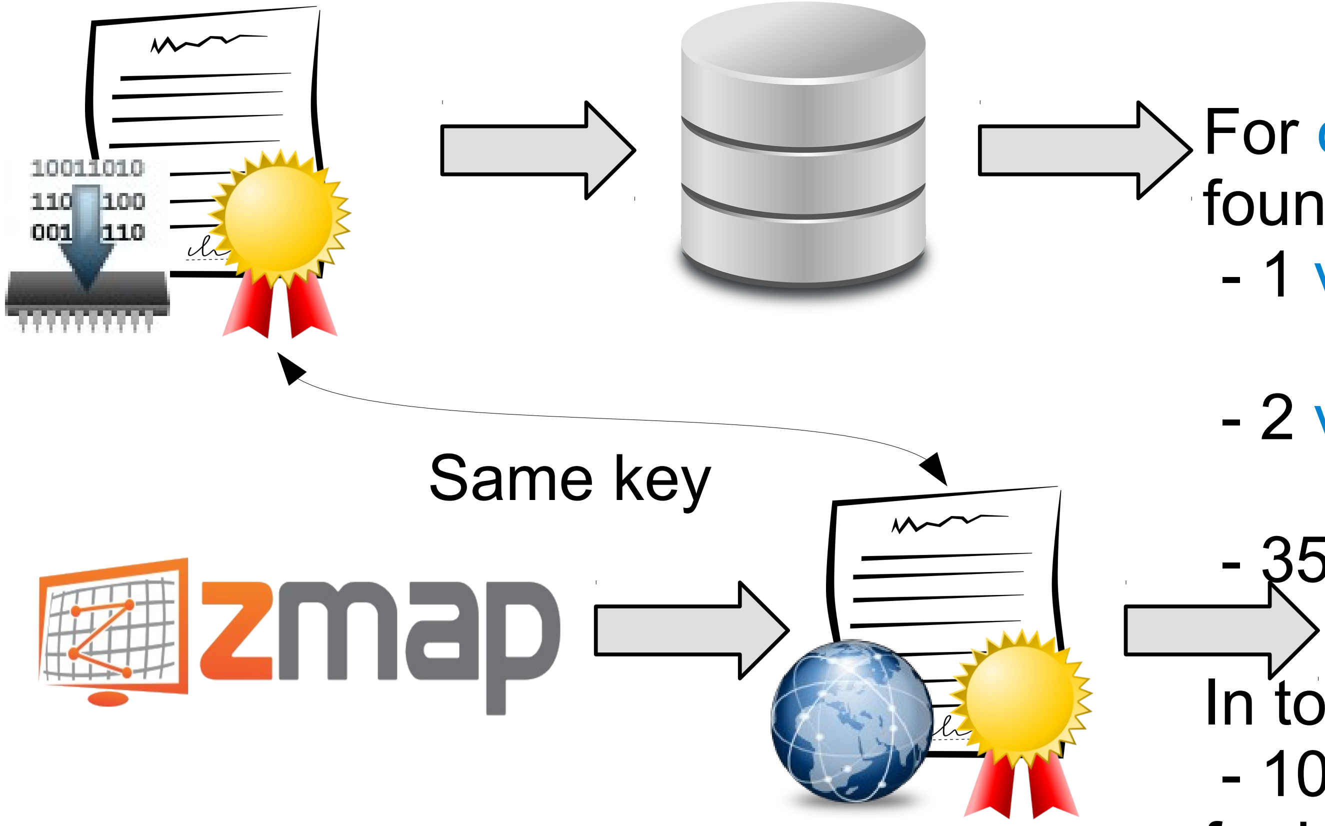
Example:
Firmware HTTPS keys correlation



Example: Firmware HTTPS keys correlation



Example:
Firmware HTTPS keys correlation



For one certificate, we found at least:

- 1 vulnerability
- 2 vendors
- 35K online devices

In total:

- 109 private RSA keys for HTTPS certificates

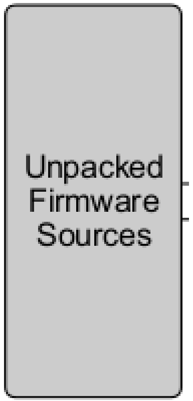
Static Firmware Analysis
Some Results

- 38 new vulnerabilities
- 693 firmware images with at least one vulnerability
- 140K online devices correlated to some vulnerabilities

Large Scale Challenge 3: Automated Dynamic Analysis

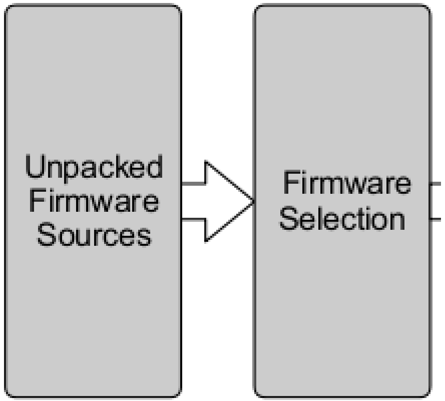
firmware · re

Dynamic Firmware Analysis Automated and Large Scale



Unpacked Firmware Sources

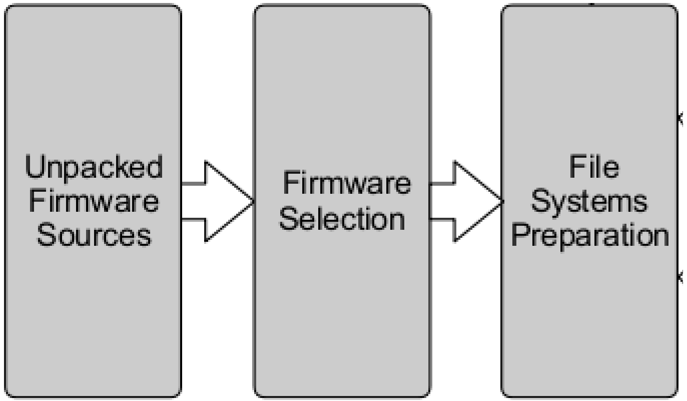
Dynamic Firmware Analysis Automated and Large Scale



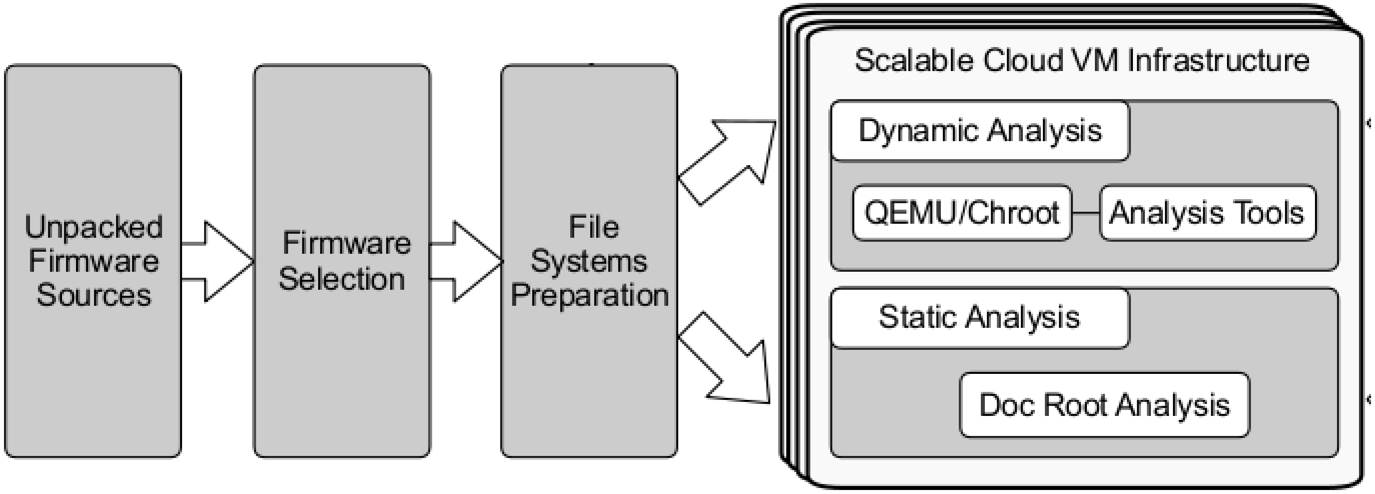
Unpacked Firmware Sources

Firmware Selection

Dynamic Firmware Analysis Automated and Large Scale



Dynamic Firmware Analysis Automated and Large Scale



Unpacked Firmware Sources

Firmware Selection

File Systems Preparation

Scalable Cloud VM Infrastructure

Dynamic Analysis

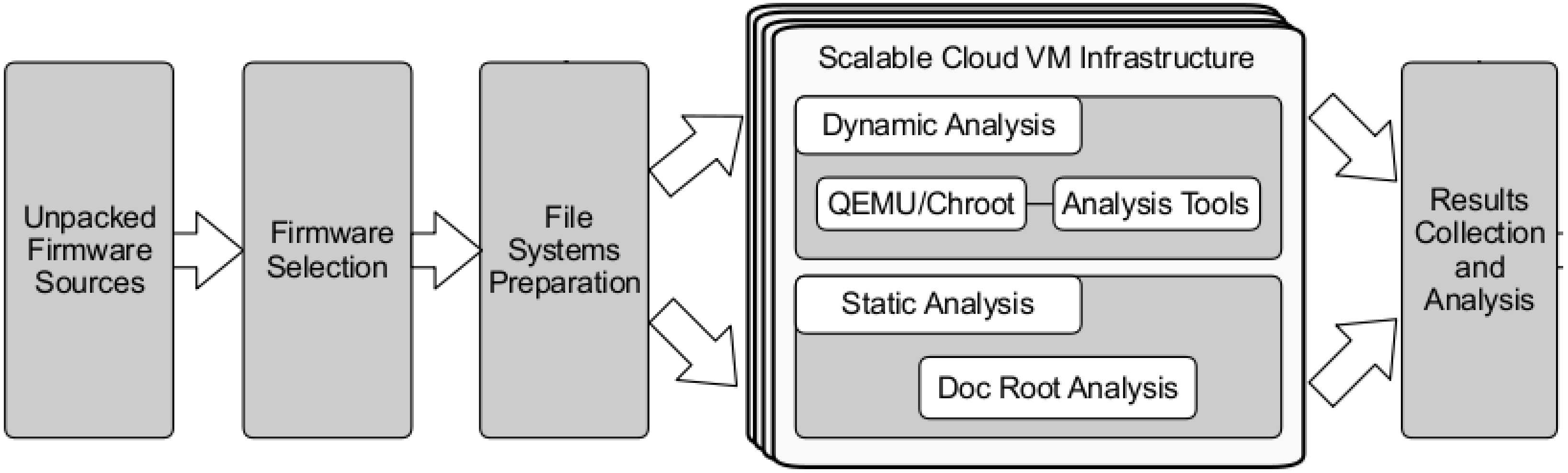
QEMU/Chroot

Analysis Tools

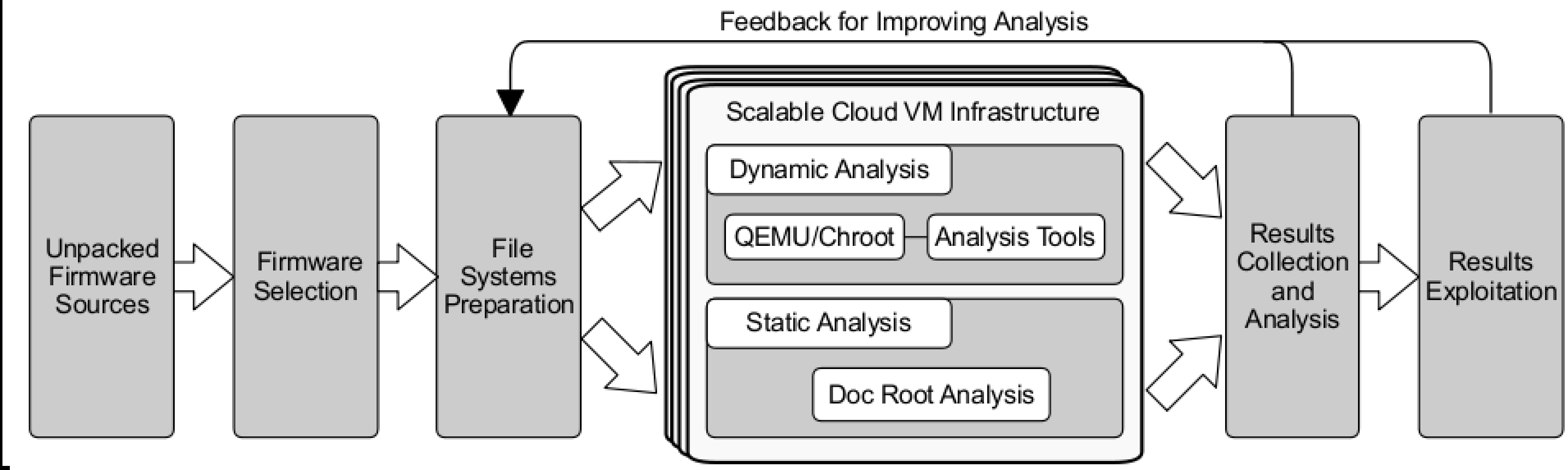
Static Analysis

Doc Root Analysis

Dynamic Firmware Analysis Automated and Large Scale

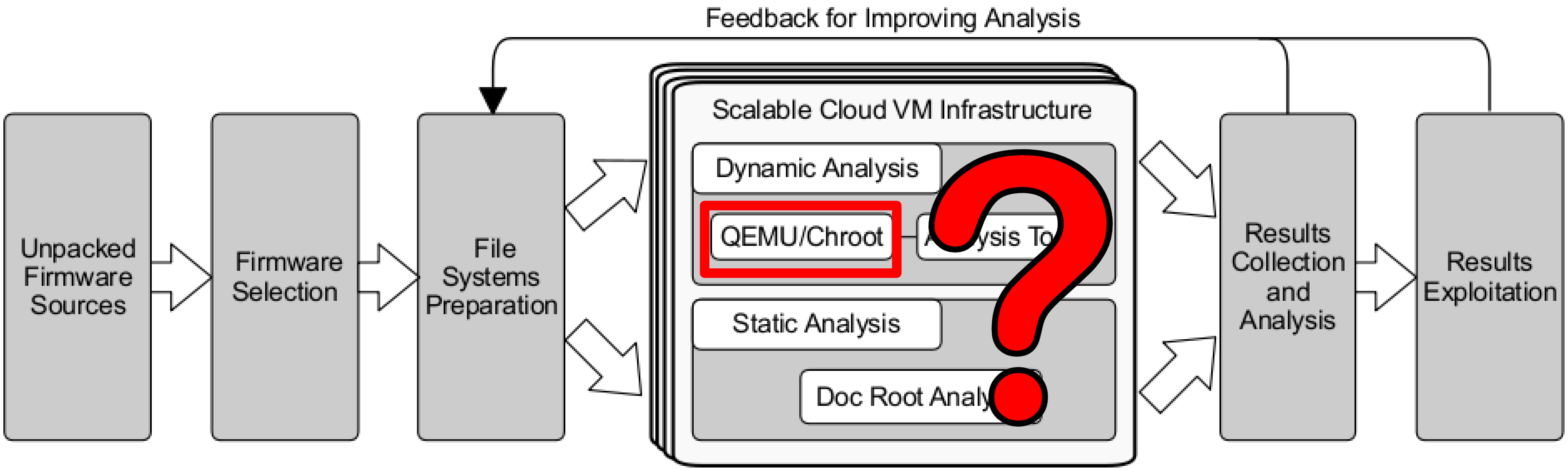


Dynamic Firmware Analysis Automated and Large Scale

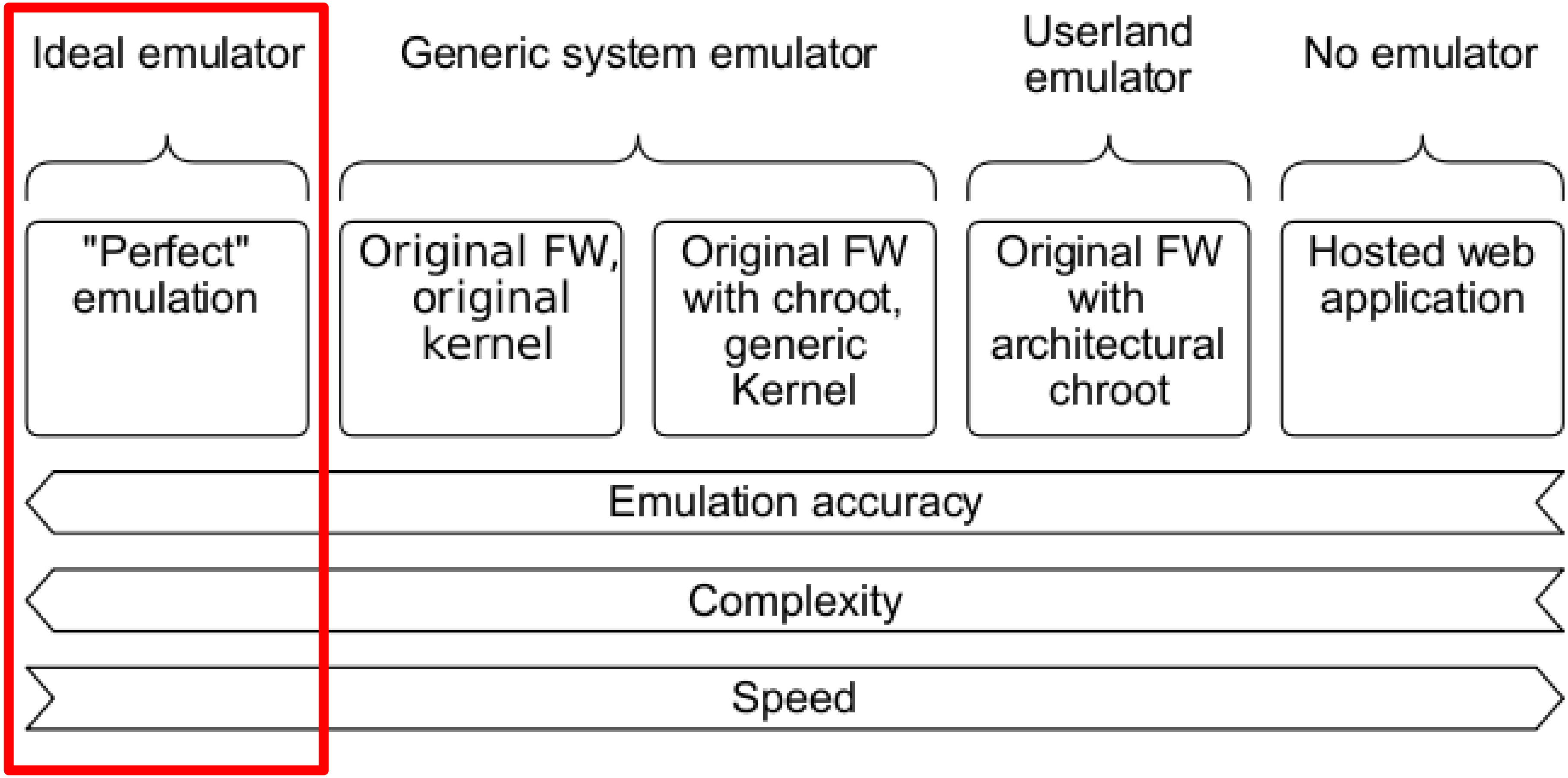


firmware · re

Dynamic Firmware Analysis Automated and Large Scale



Dynamic Firmware Analysis
Emulator's Dilemma



Ideal emulator

Generic system emulator

Userland emulator

No emulator

"Perfect" emulation

Original FW, original kernel

Original FW with chroot, generic Kernel

Original FW with architectural chroot

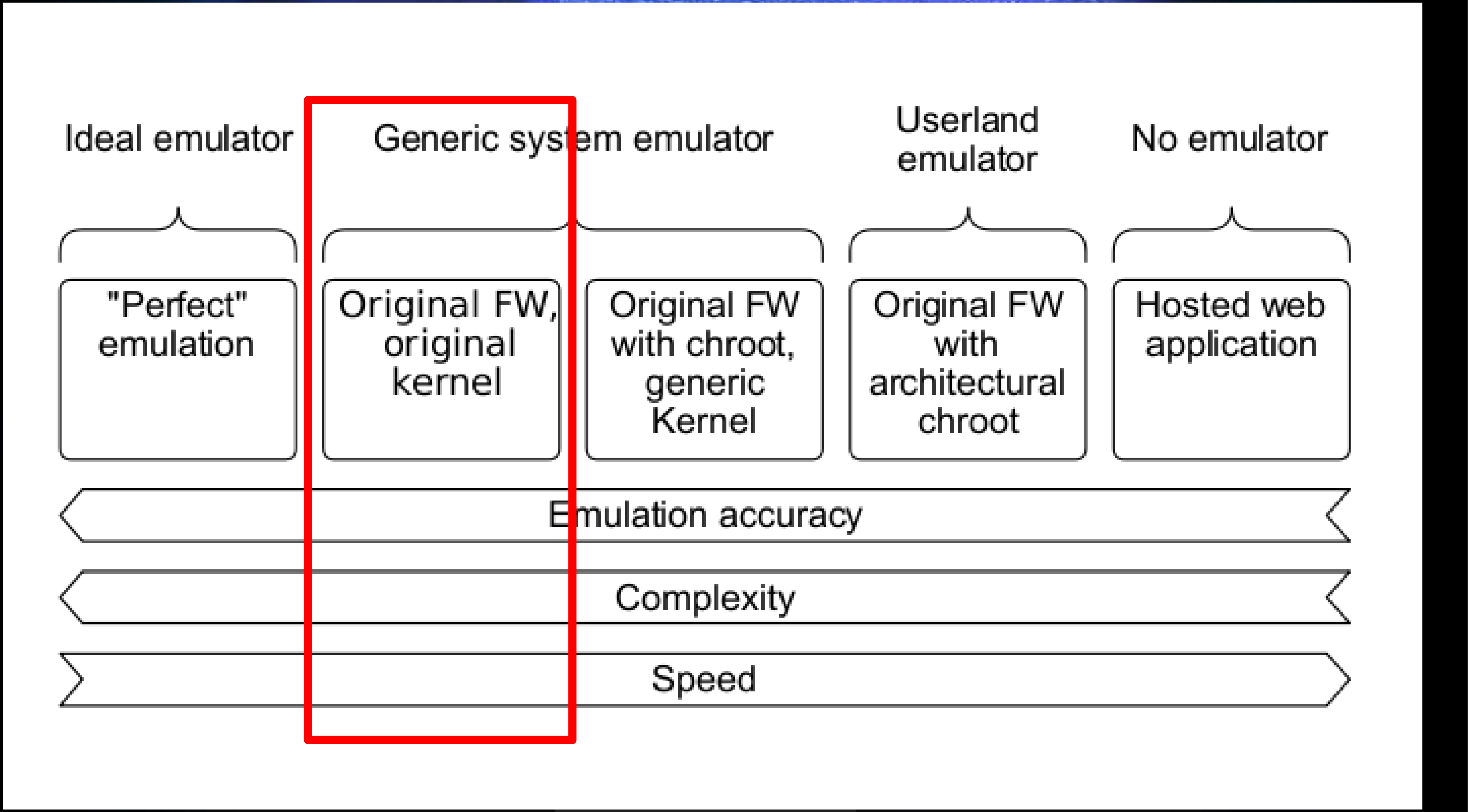
Hosted web application

Emulation accuracy

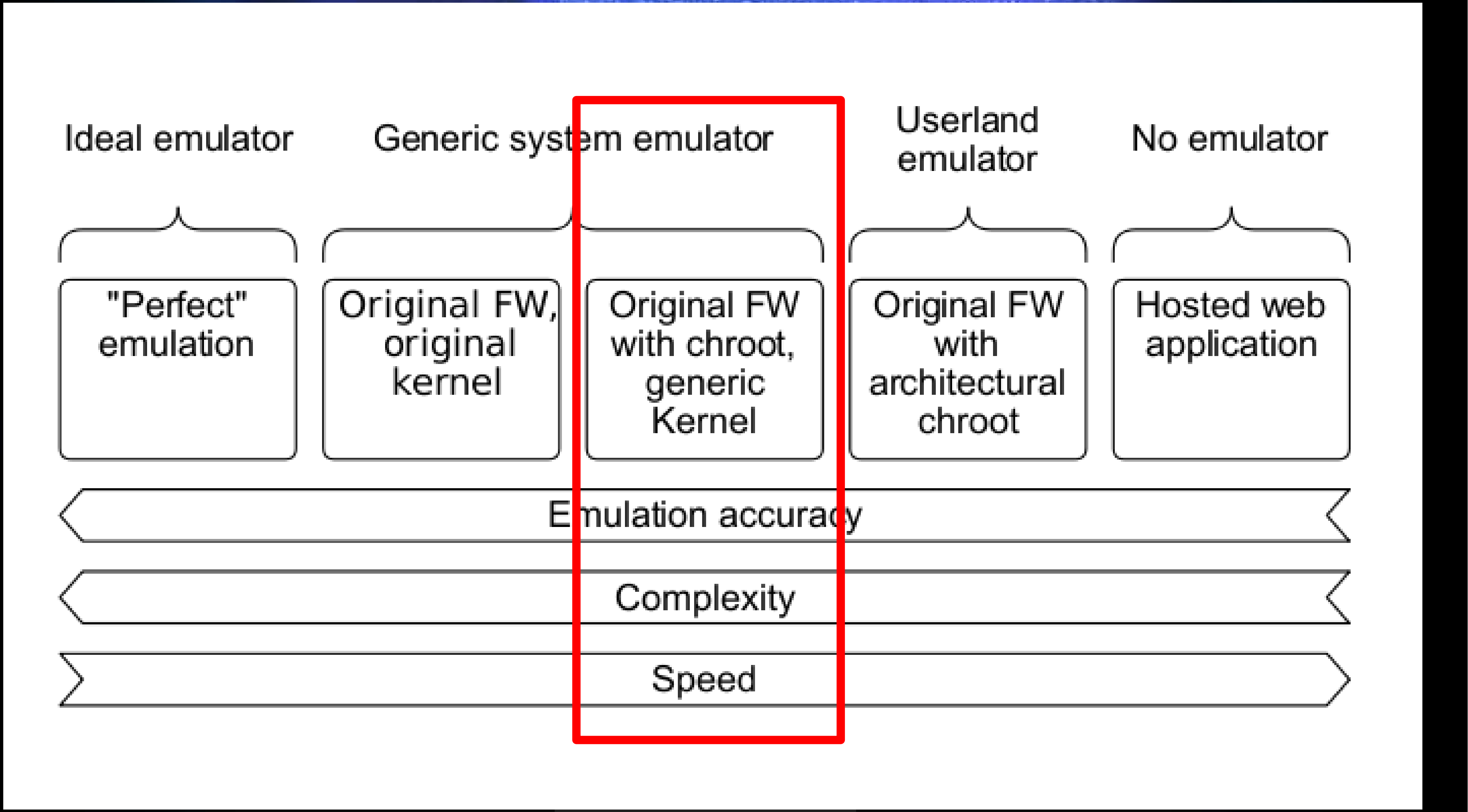
Complexity

Speed

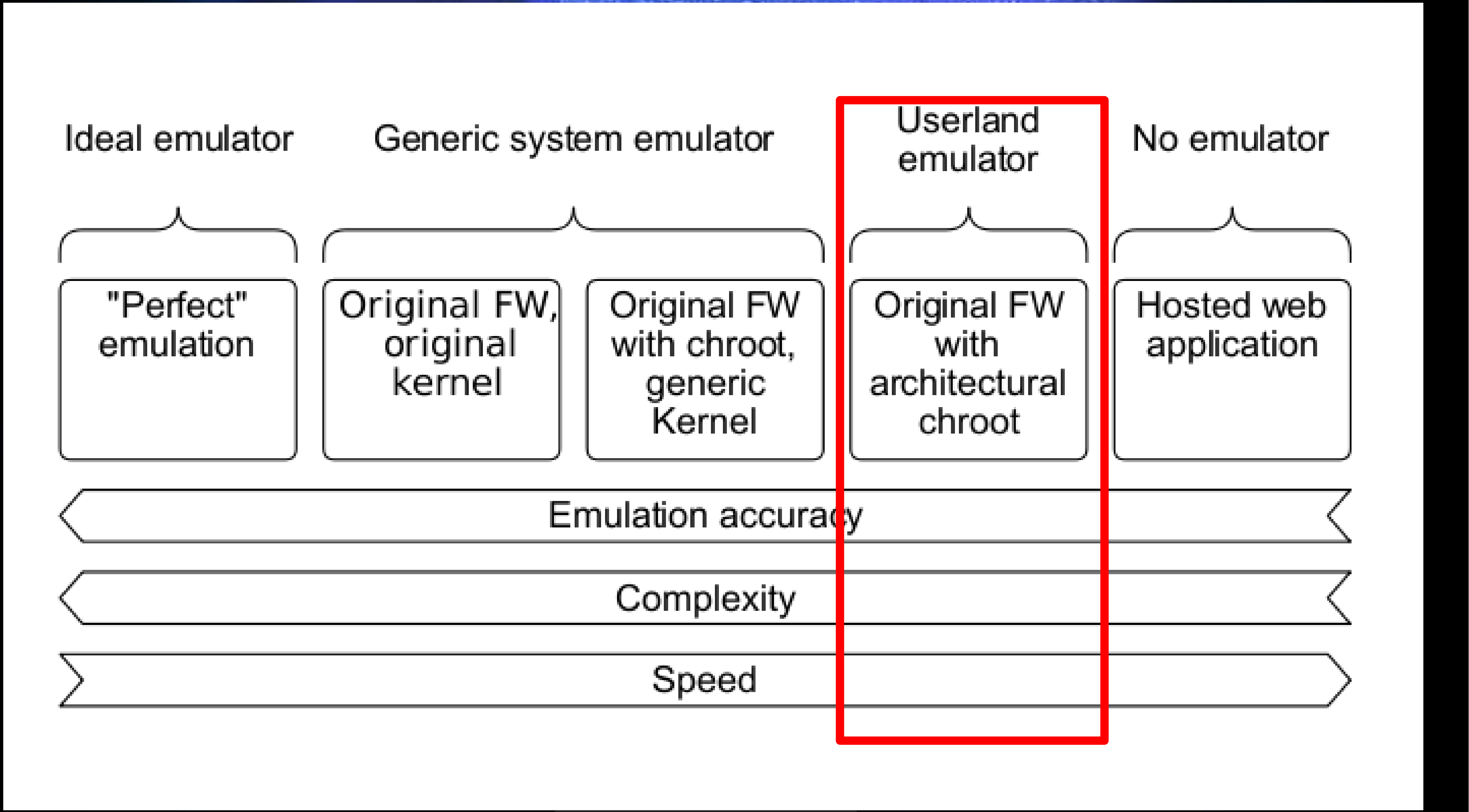
Dynamic Firmware Analysis
Emulator's Dilemma



Dynamic Firmware Analysis
Emulator's Dilemma

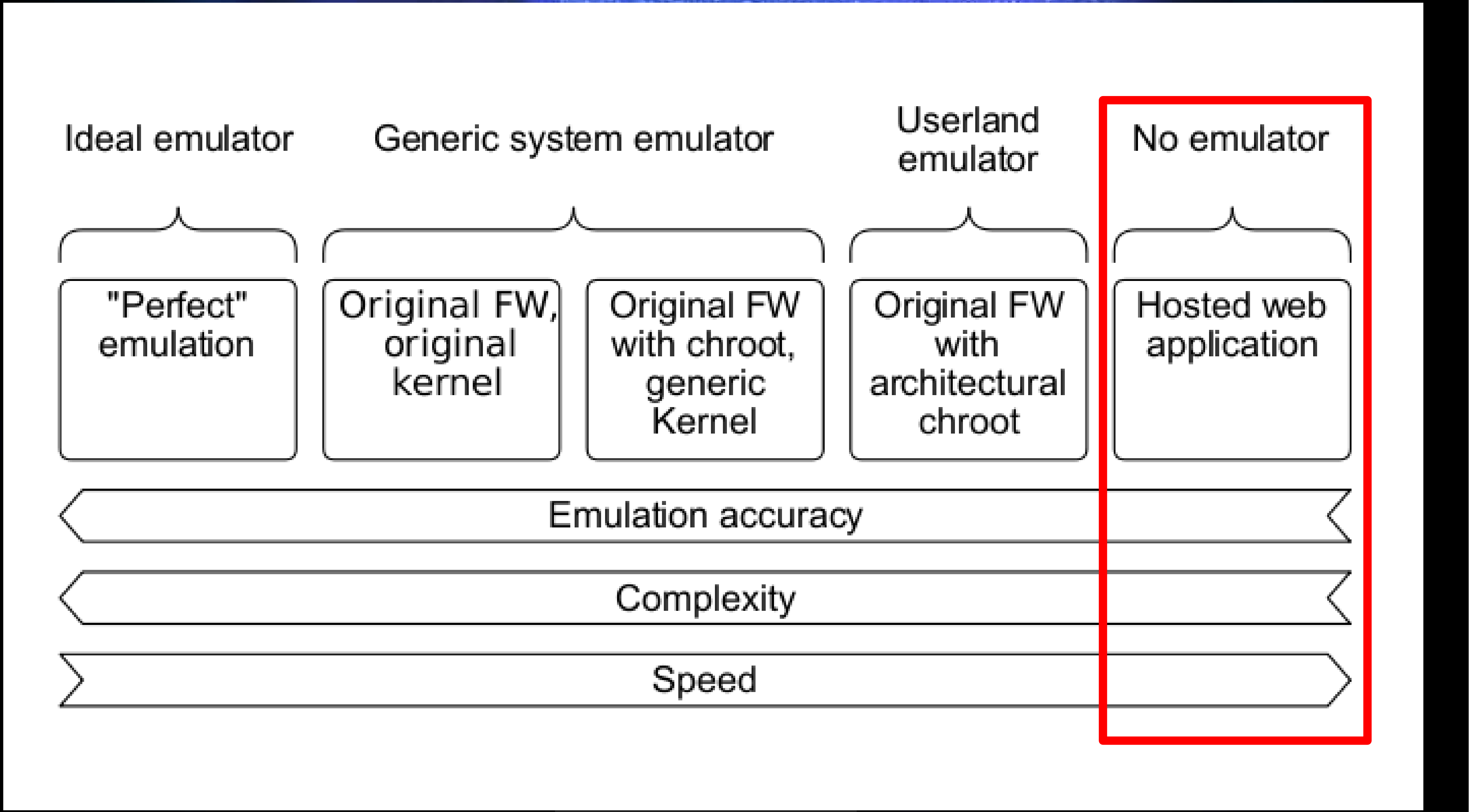


Dynamic Firmware Analysis Emulator's Dilemma

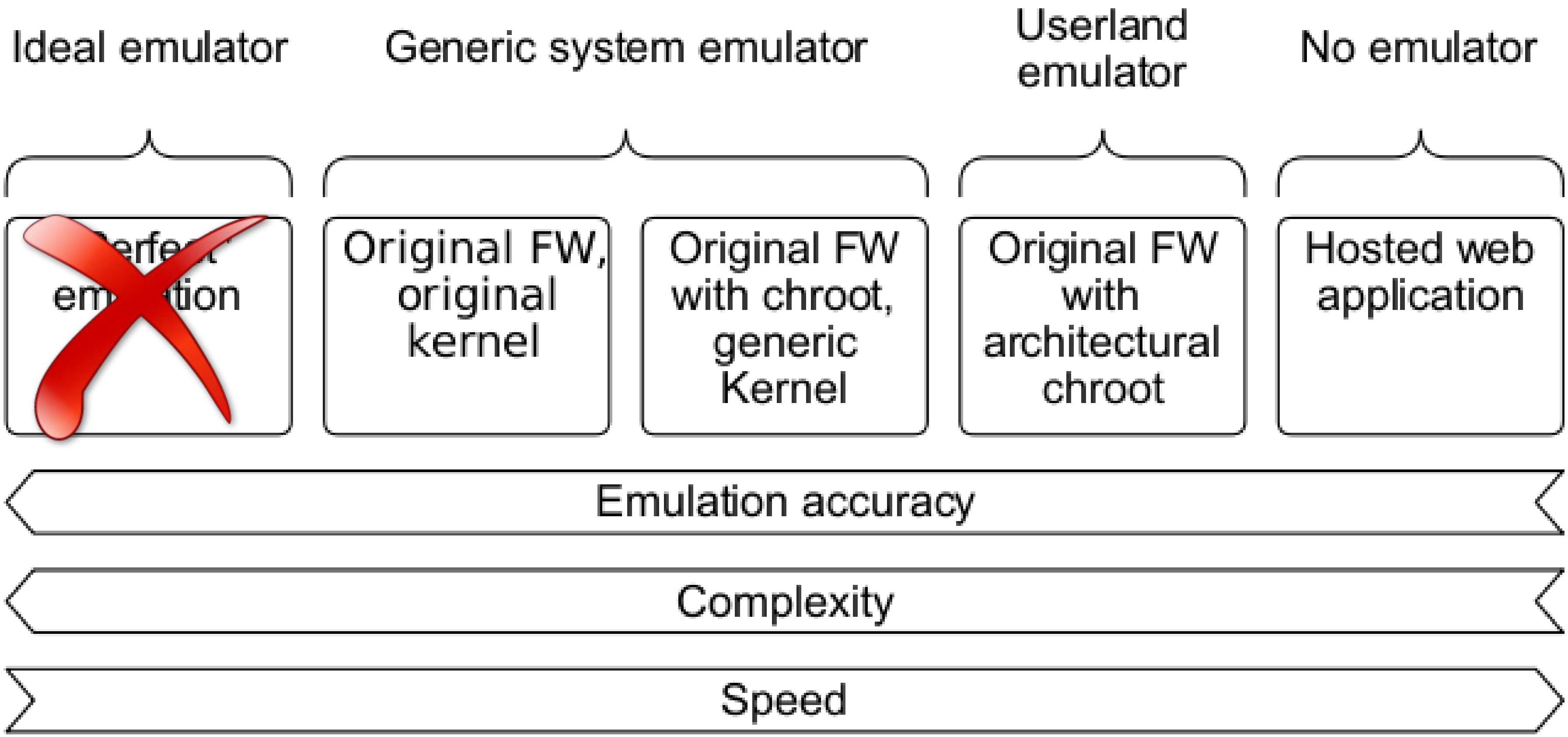


firmware · ǝɹ

Dynamic Firmware Analysis
Emulator's Dilemma



Dynamic Firmware Analysis
Emulator's Dilemma



Ideal emulator

Generic system emulator

Userland emulator

No emulator

Perfect emulation

Original FW, original kernel

Original FW with chroot, generic Kernel

Original FW with architectural chroot

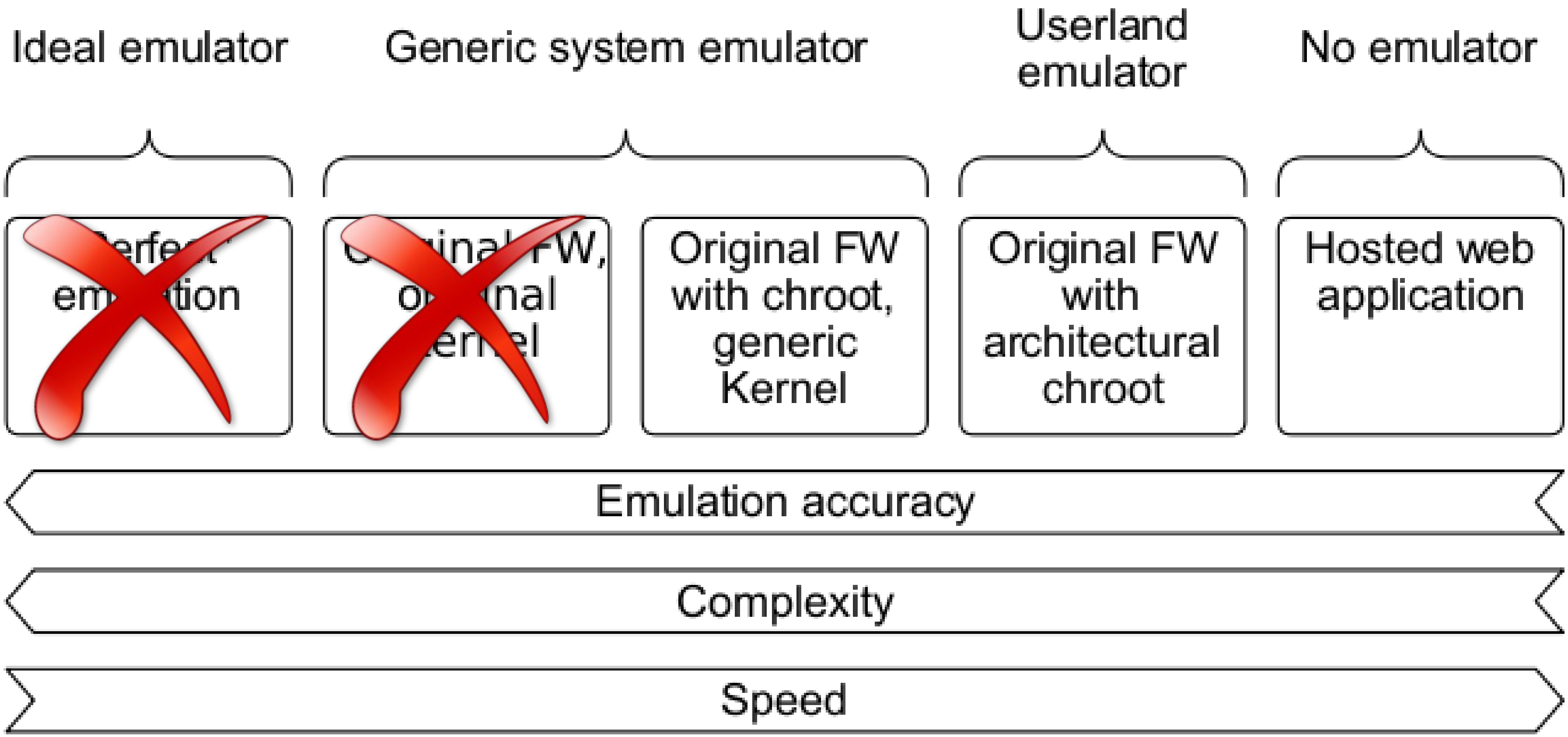
Hosted web application

Emulation accuracy

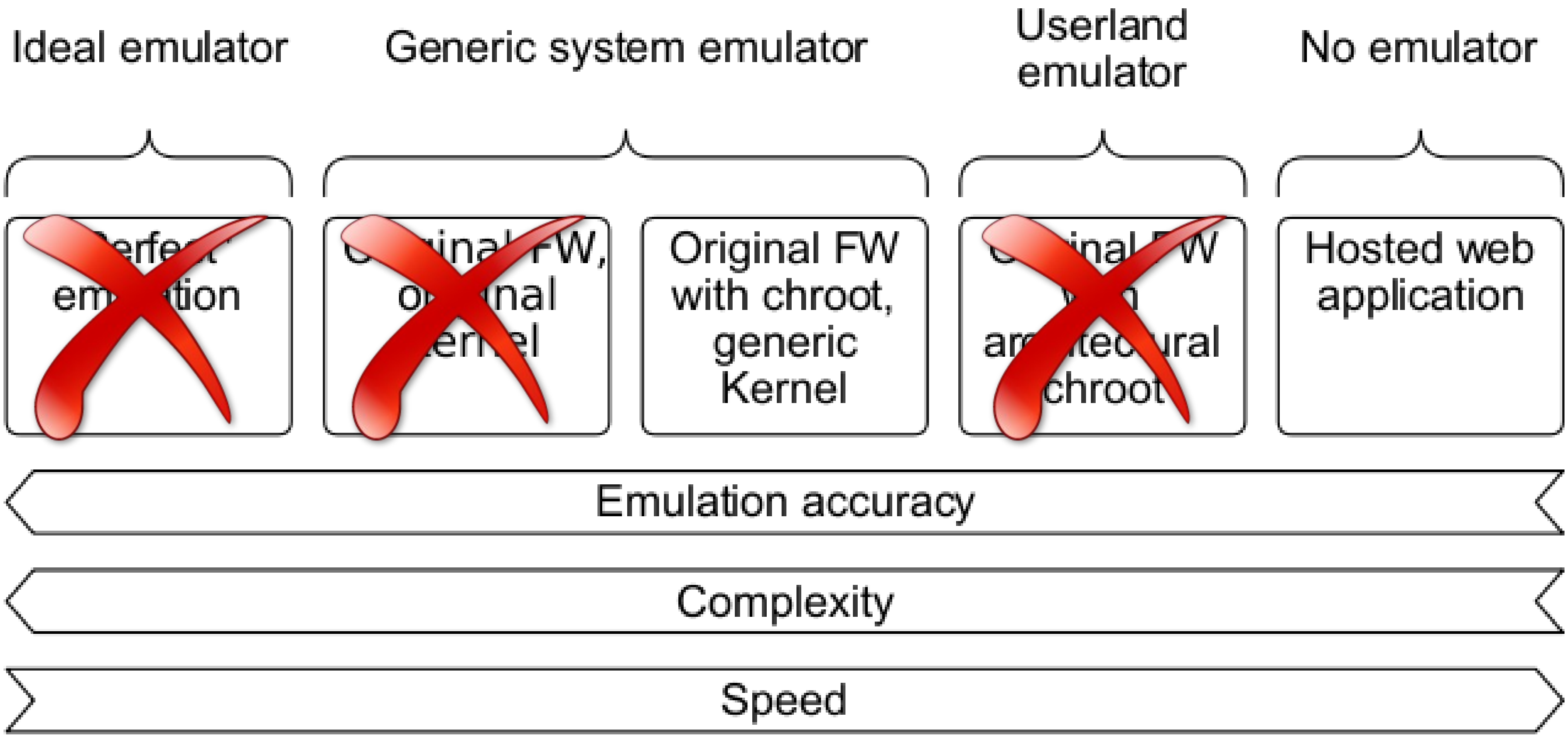
Complexity

Speed

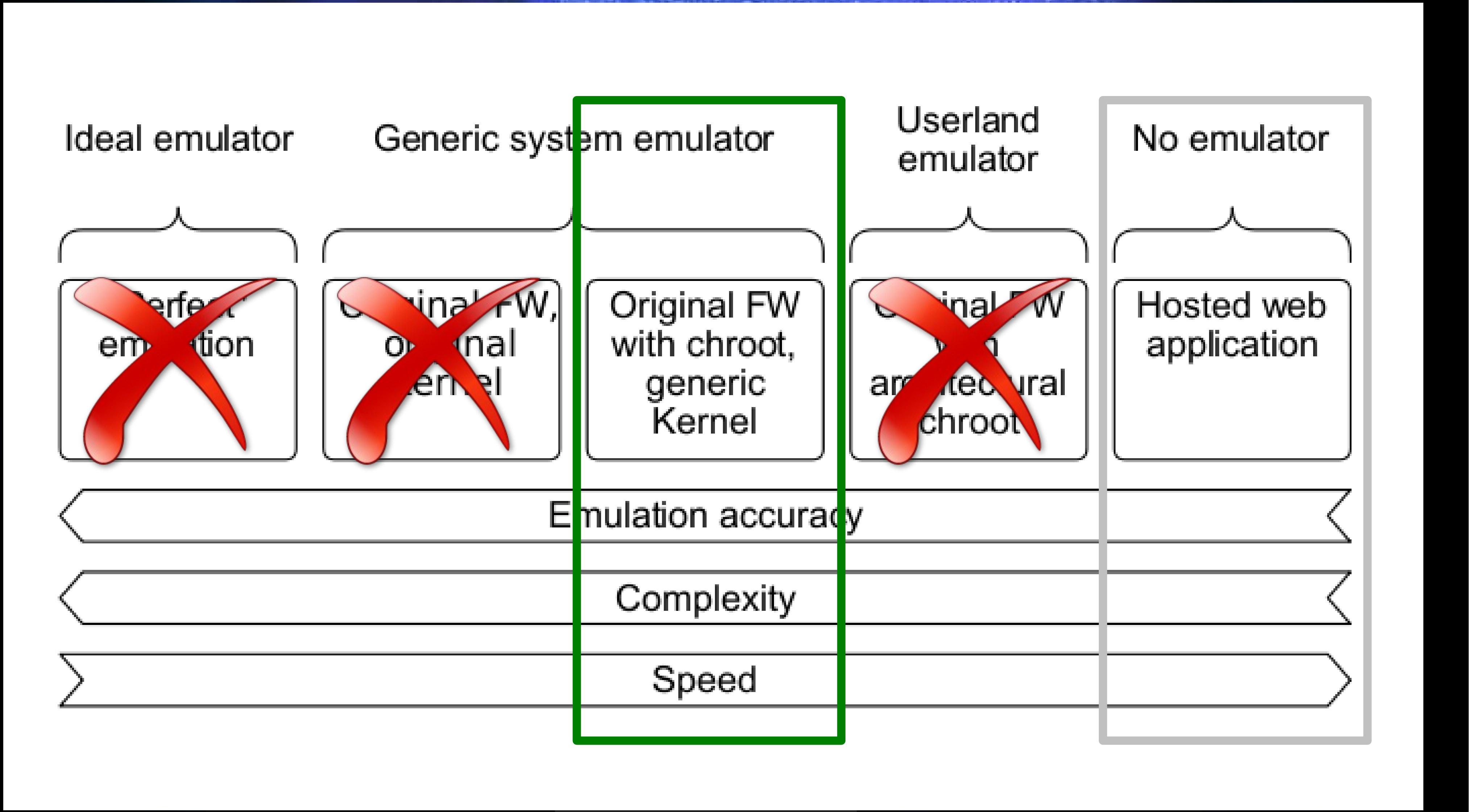
Dynamic Firmware Analysis
Emulator's Dilemma



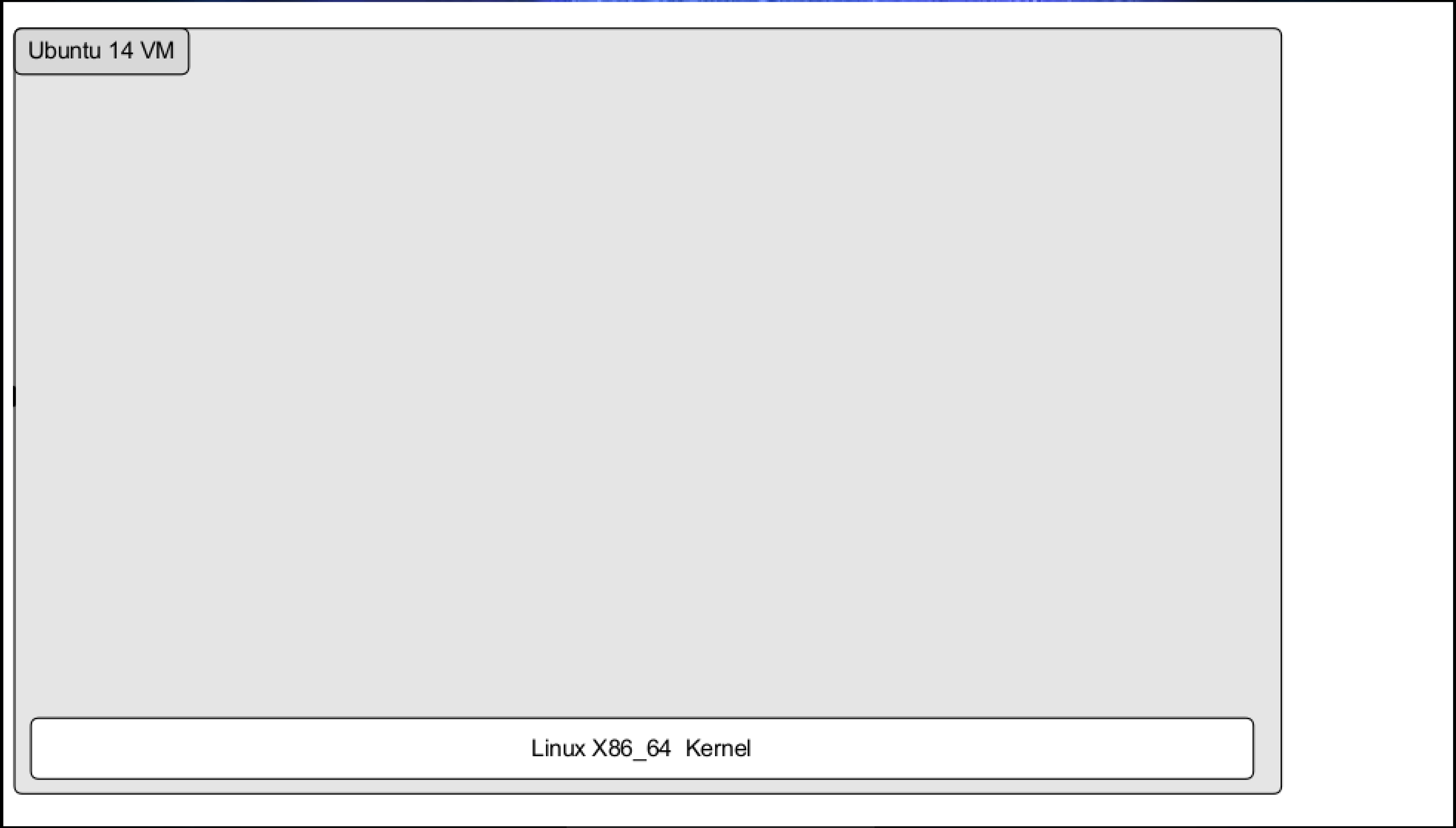
Dynamic Firmware Analysis
Emulator's Dilemma



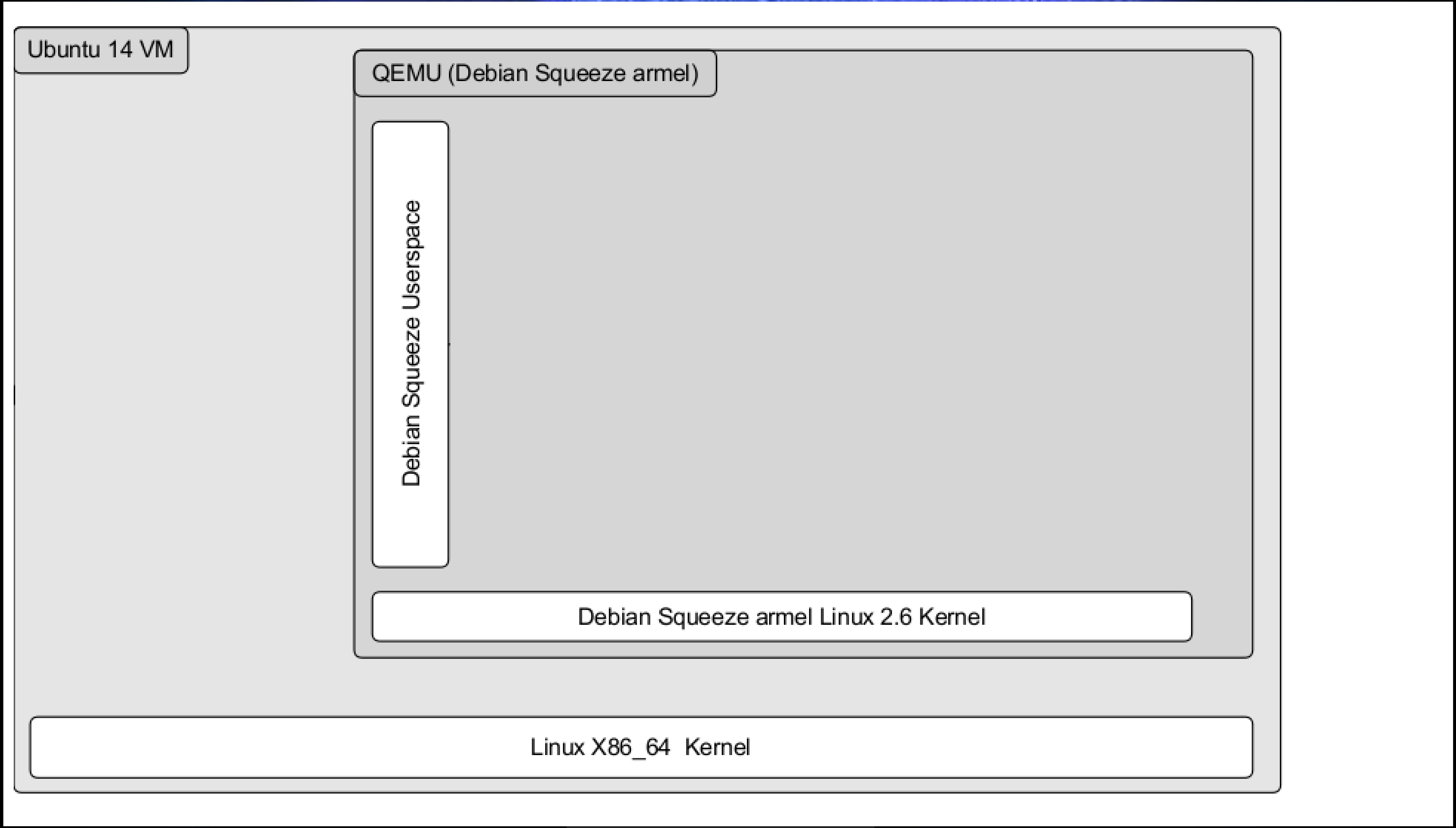
Dynamic Firmware Analysis
Emulator's Dilemma



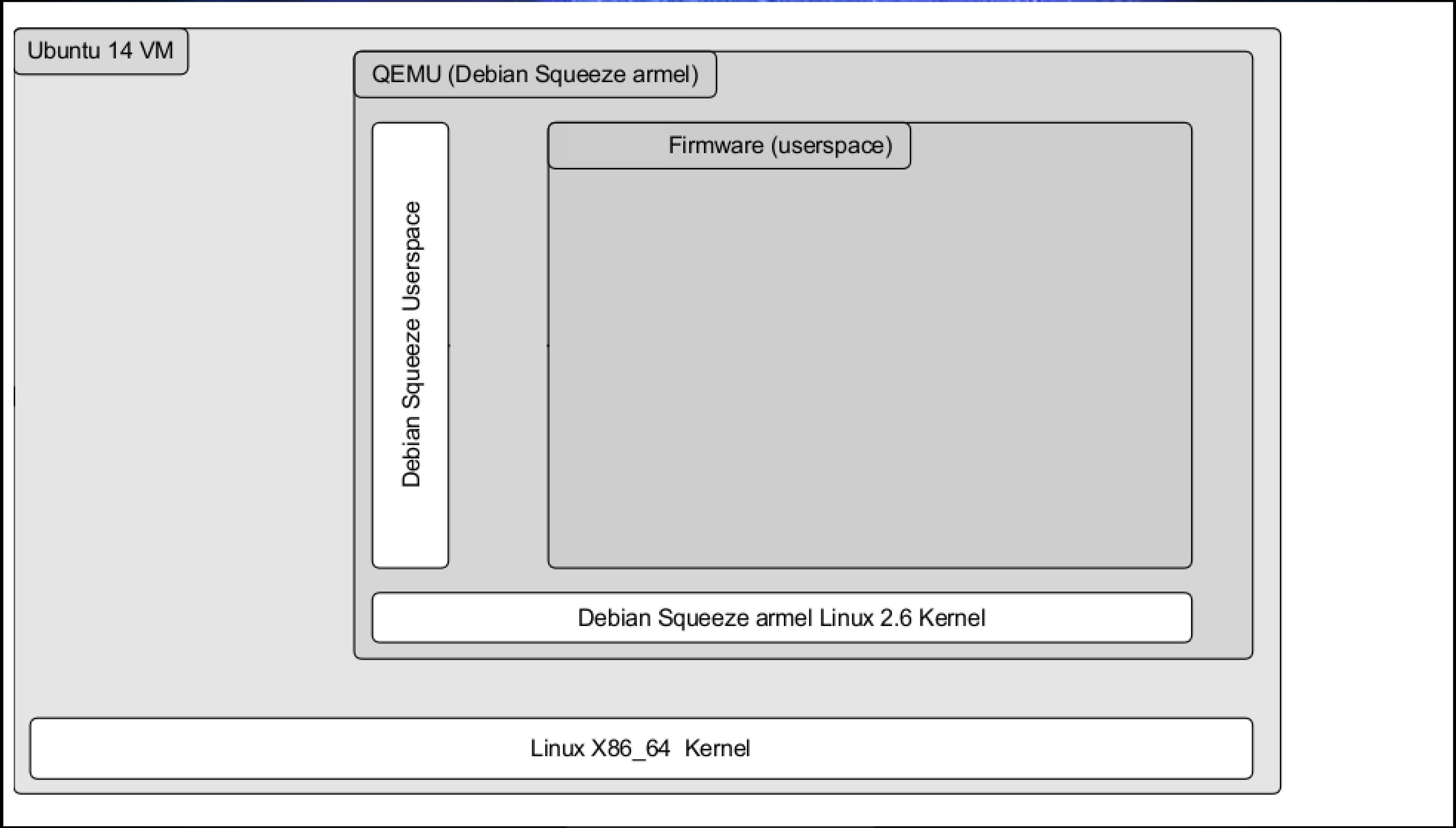
Dynamic Firmware Analysis
Scalable Emulation and Analysis



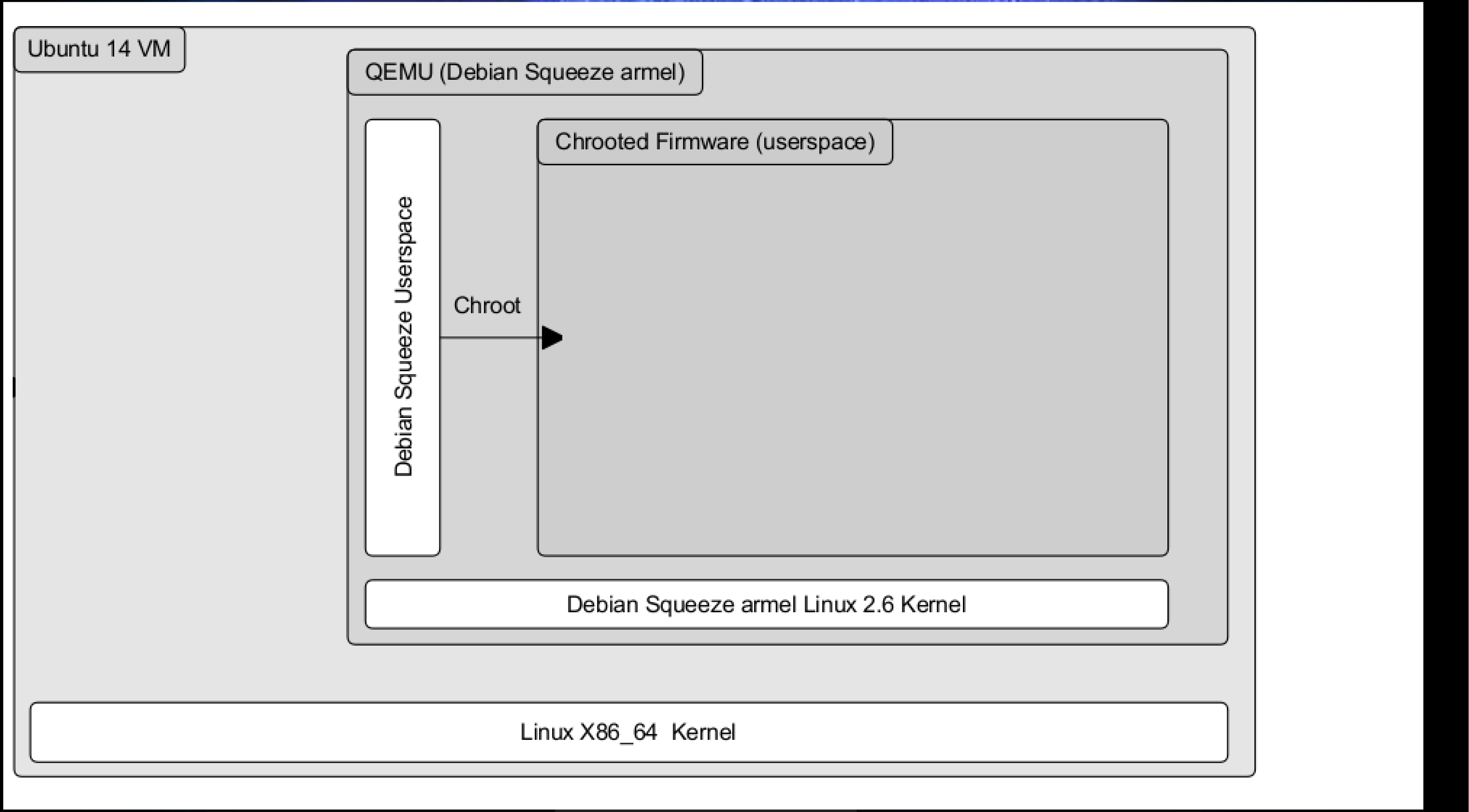
Dynamic Firmware Analysis
Scalable Emulation and Analysis


Ubuntu 14 VM
QEMU (Debian Squeeze armel)
Debian Squeeze Userspace
Debian Squeeze armel Linux 2.6 Kernel
Linux X86_64 Kernel

Dynamic Firmware Analysis
Scalable Emulation and Analysis

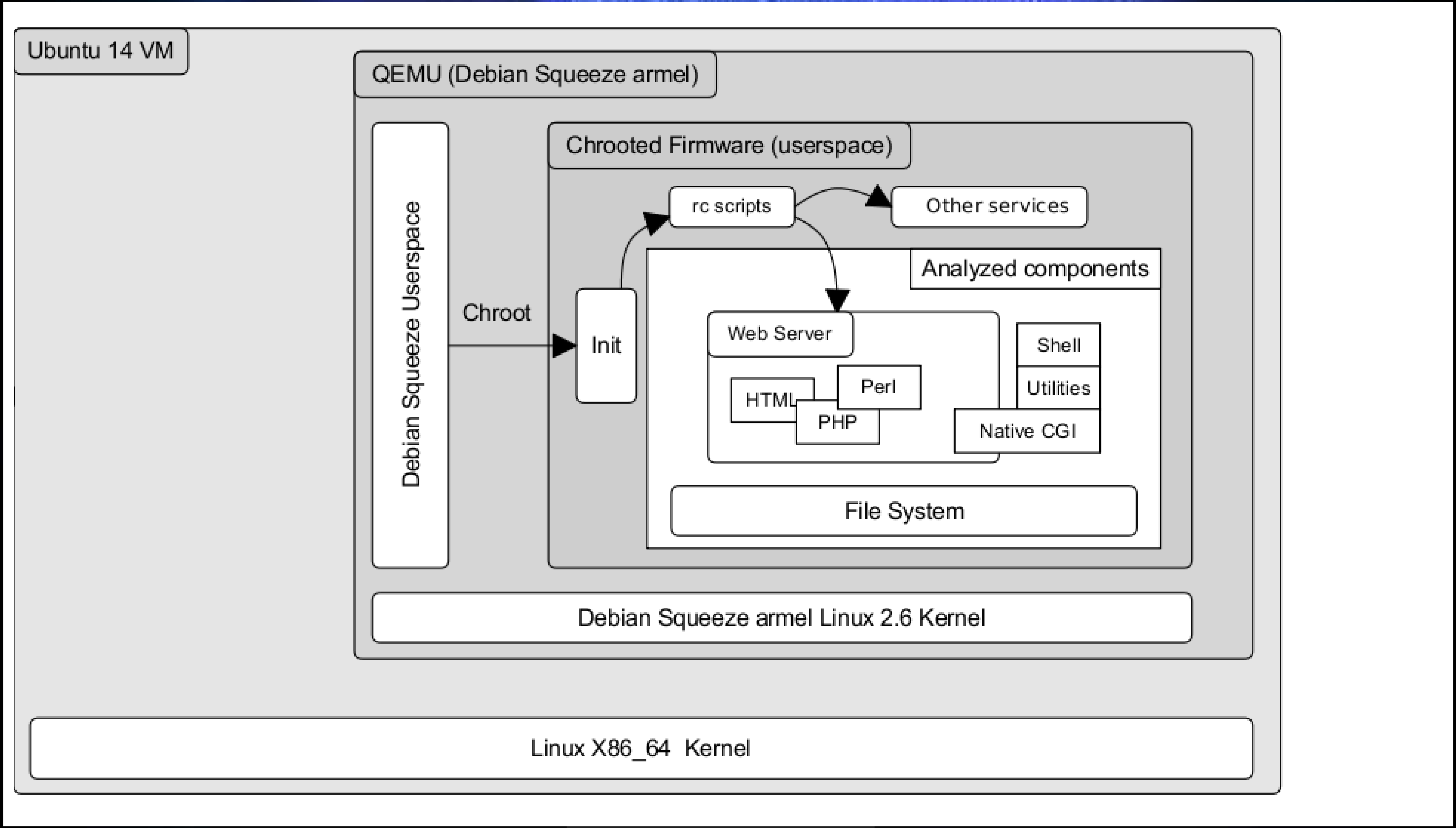


Dynamic Firmware Analysis
Scalable Emulation and Analysis

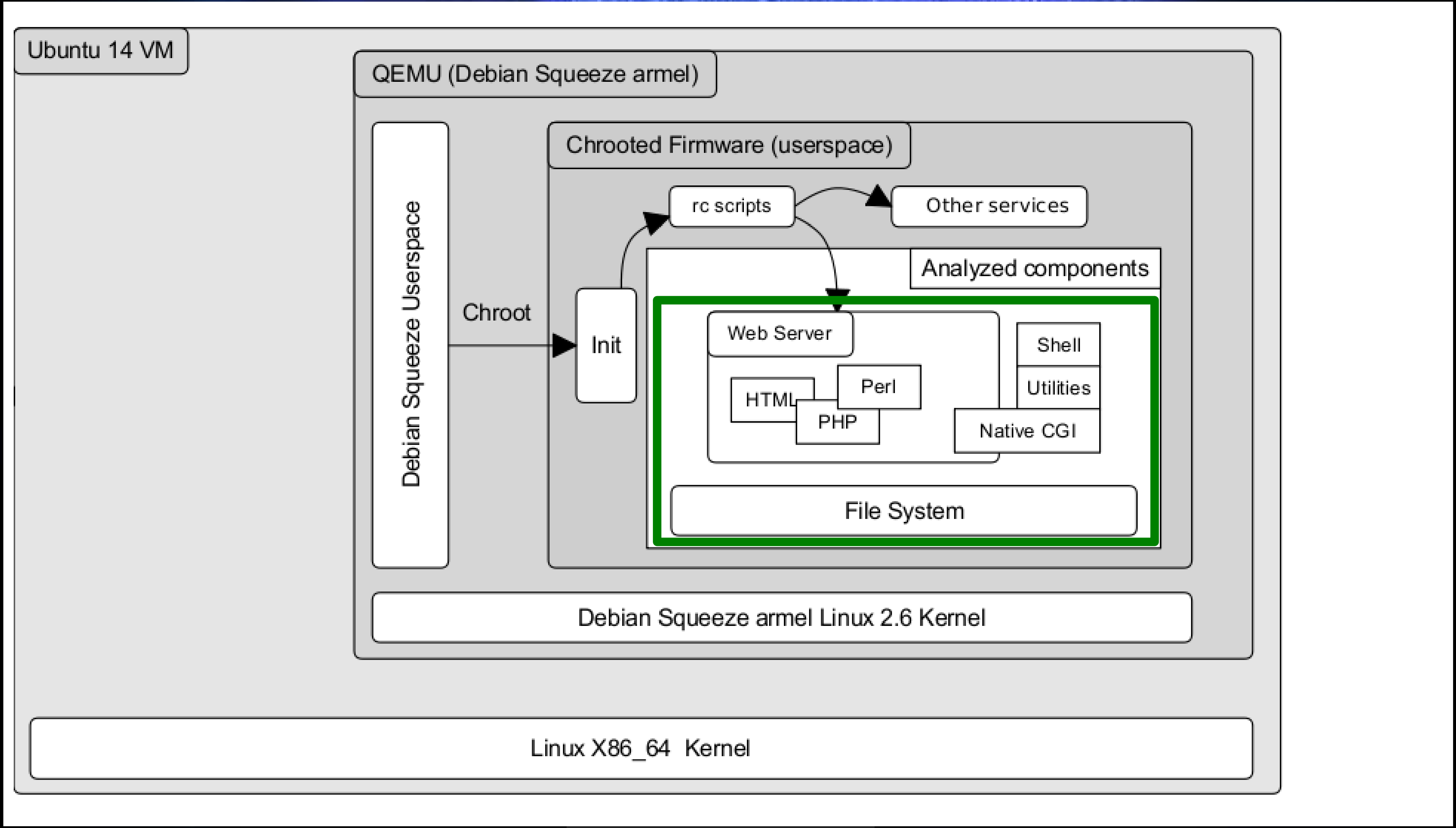


Ubuntu 14 VM
QEMU (Debian Squeeze armel)
Debian Squeeze Userspace
Chroot
Chrooted Firmware (userspace)
Debian Squeeze armel Linux 2.6 Kernel
Linux X86_64 Kernel

Dynamic Firmware Analysis Scalable Emulation and Analysis

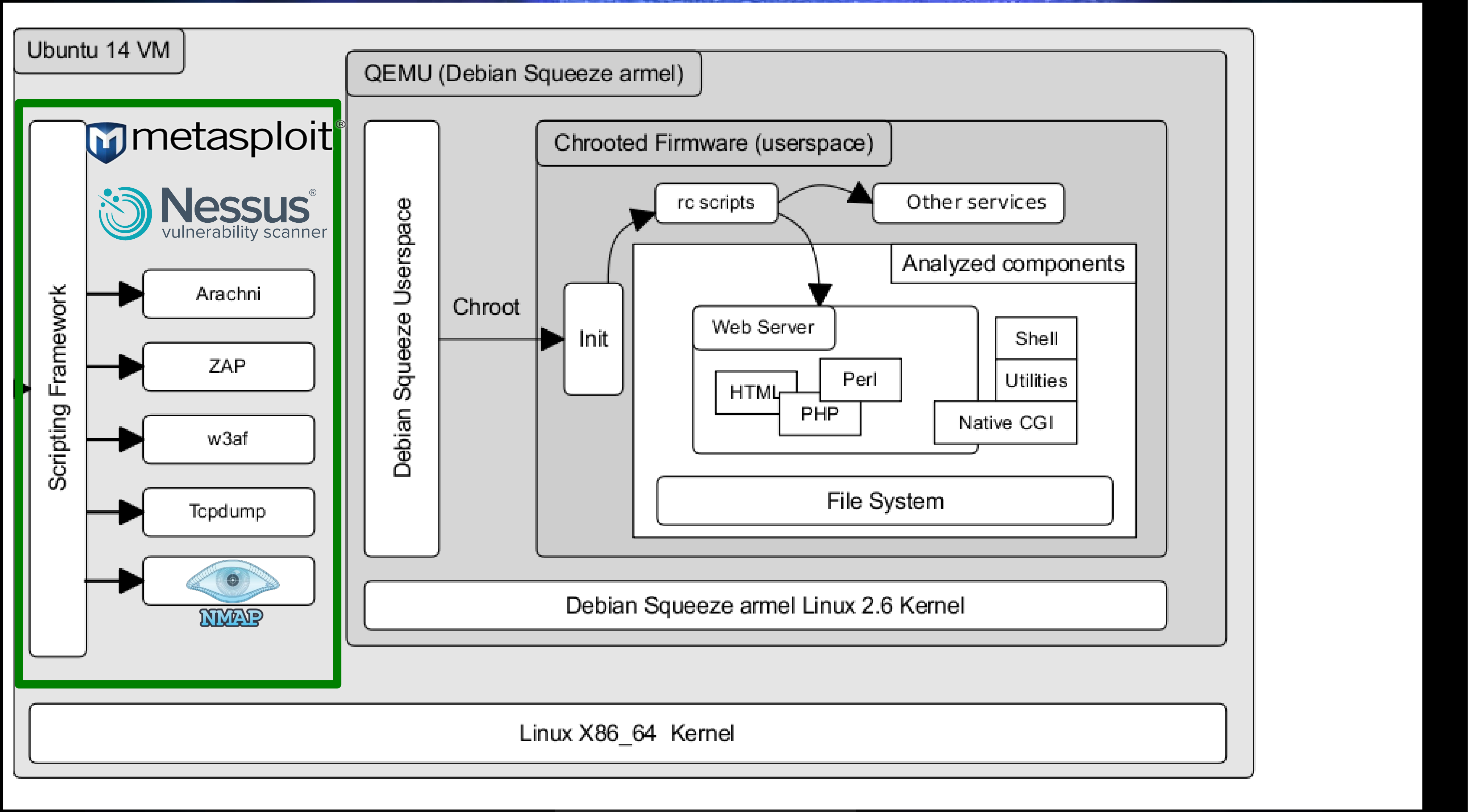


Dynamic Firmware Analysis Scalable Emulation and Analysis



firmware · ǝɹ

Dynamic Firmware Analysis
Scalable Emulation and Analysis



firmware · ǝɹ

Dynamic Firmware Analysis
Some Results

- High-severity vulnerability impact
  - Command injection, XSS, CSRF
  - Automated+scalable static and dynamic analysis
  - 225 high-severity vulnerabilities, many previously unknown
  - 185 firmware images (~10% of original)
  - 13 vendors (~25% of original)
- Total alerts from the tools
  - 6068 dynamic analysis alerts on 58 firmware images
  - 9046 static analysis alerts on 145 firmware images
  - Manual triage and confirmation is challenging

Applications

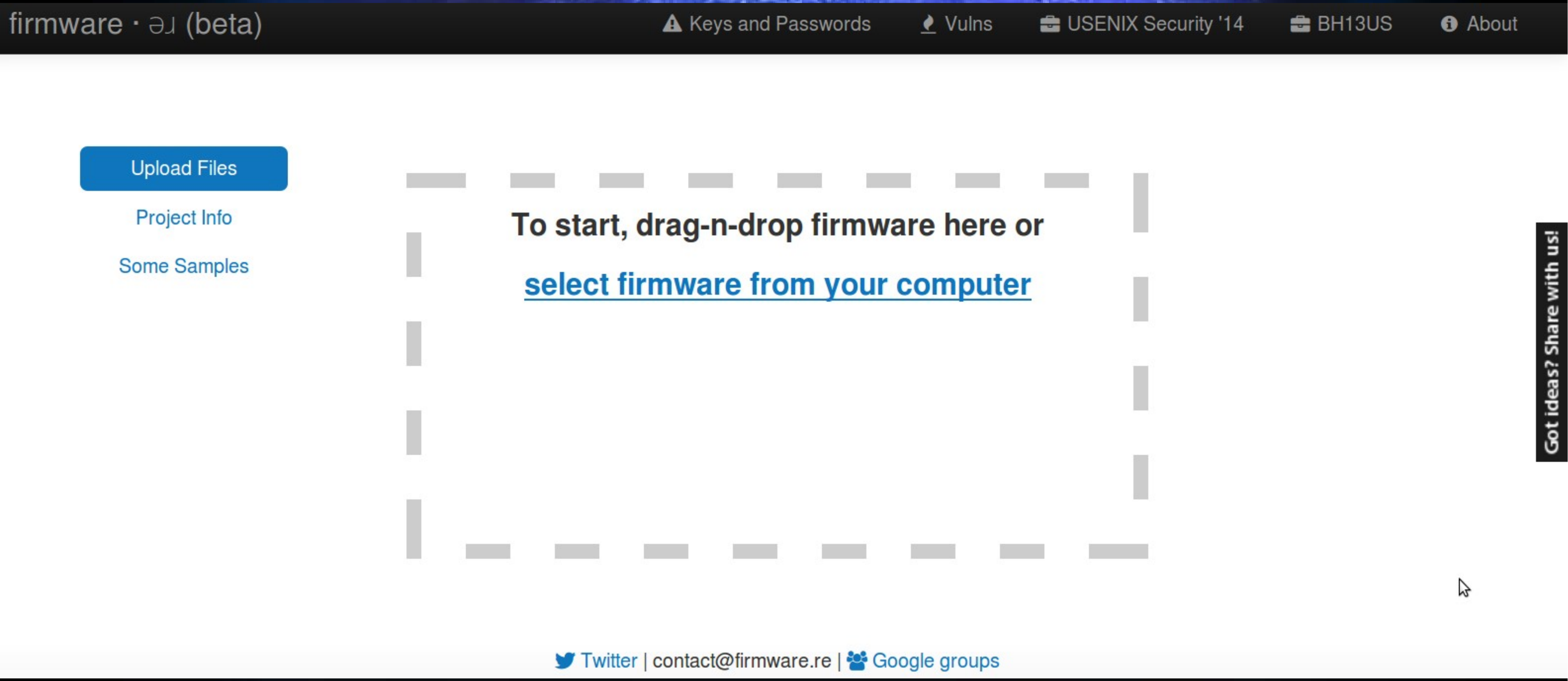
Application Example
Industry Players

- 1 big player in SCADA/ICS/embedded
  - In ”Top 100” of ”Fortune Global 500” (2015)

- 3 years R&D contract (from 2015)

- Using our frameworks
  - For their own firmware life-cycle
  - Firmware collection, unpacking, analysis
  - Dynamic analysis and symbolic execution

Firmware.RE
First project of its kind



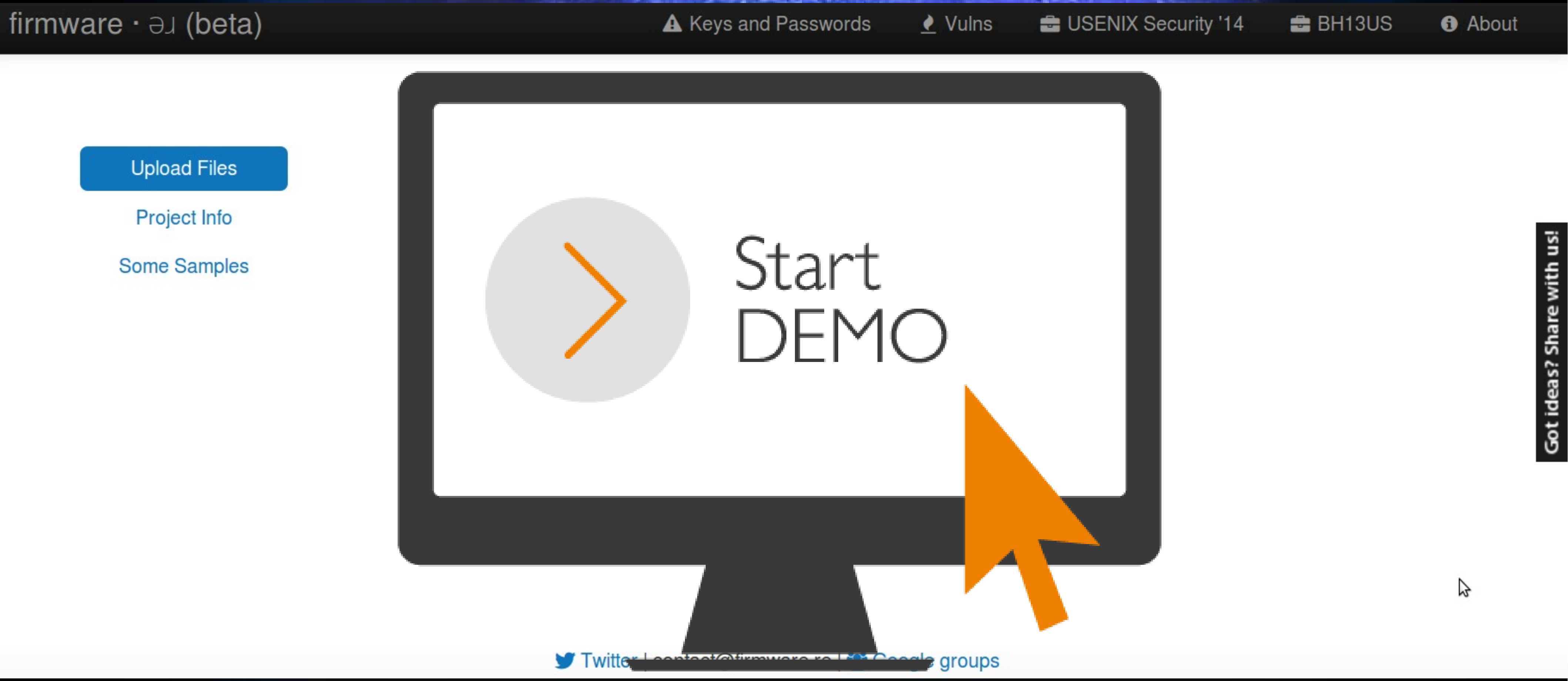
To start, drag-n-drop firmware here or

select firmware from your computer

Twitter | contact@firmware.re | Google groups

firmware · əɹ

Firmware.RE Demo Time!



firmware · ǝɹ

Conclusions

- Plenty of latent vulnerabilities in embedded firmware
- Firmware security analysis is absolutely necessary
- Involves many untrivial steps and challenges
- A broader view on firmwares is not just beneficial, but necessary
- Security
  - Tradeoff with both cost and time-to-market
  - Clearly not a priority for some vendors

Summary

- We build-up research expertise and implement our expertise in working prototypes
- First framework for automated large scale security analysis and classification of firmwares and embedded devices
  - Simple and advanced analysis using dynamic and static
  - Quick identification of (un)known vulnerabilities
  - Automated classification and fingerprinting

References

- www.firmware.re
- www.s3.eurecom.fr/~costin/

Collaborators Acknowledgements & Thanks

- Dr. Jonas Zaddach
- Prof. Aurelien Francillon
- Prof. Davide Balzarotti
- Dr. Apostolis Zarras

The End

Thank You!

Questions?

{name}@firmware.re